Mechanism Constraints and Singularities - The Algebraic Formulation

Manfred L. Husty and Dominic R. Walter
University of Innsbruck, Austria

Abstract Various mathematical formulations are used to describe mechanism and robot kinematics. This mathematical formulation is the basis for kinematic analysis and synthesis, i.e., determining displacements, velocities and accelerations, on the one hand, and obtaining design parameters on the other. Vector/matrix formulation containing trigonometric functions is arguably the most favoured approach used in the engineering research community. A less well known but nevertheless very successful approach relies on an algebraic formulation. This involves describing mechanism constraints with algebraic (polynomial) equations and solving these equation sets, that pertain to some given mechanism or robot, with the powerful tools of algebraic and *numerical* algebraic geometries.

In the first section of this chapter the algebraic formulation of Euclidean displacements using Study coordinates (or dual quaternions) will be recalled. Then it will be shown how constraint equations of different kinematic chains can be derived using either geometric insight, elimination methods or the linear implicitization algorithm (LIA). LIA will be described in detail because it can be used even without deep kinematic and geometric insight into the properties of the kinematic chain. In case the kinematic chain consists of only elementary joints the constraint equations consist of a set of polynomial equations. In the language of algebraic geometry the corresponding polynomials form an ideal which can be treated with almost classical methods, that might not be well known in the engineering community. Therefore in the third section these methods will be recalled and it will be shown how they can be used to derive the properties of the kinematic entities. In the last section we apply the introduced algebraic methods to the complete analysis of the 3-UPU-TSAI parallel manipulator.

1 Introduction

The mathematical description of mechanisms or robots needs essentially two ingredients: the description of Euclidean displacements and the mathematical description of the mechanical device itself. The literature (e.g. Bottema and Roth (1979)) knows many different representations of the Euclidean displacement group

$SE(3)$ such as Euler angles, Rodrigues parameters, Euler parameters, Study parameters, quaternions and dual quaternions. Within this chapter the algebraic description of mechanisms is sought and therefore Study parametrization will be used throughout. Concerning the description of the mechanisms we adopt the following point of view: No matter how complicated the mechanism might be, it consist of one or more kinematic chains. The kinematic chain is a serial assembly of joints and links and the mechanism or the robot is a serial, parallel or hybrid assembly of kinematic chains. Vector/matrix formulation containing trigonometric functions is arguably the most favoured approach used in the engineering research community. Most of the time vector loop equations are used to describe the kinematic chains and very often only a single numerical solution to the basic tasks like inverse or direct kinematics is obtained. Complete analysis and synthesis of mechanisms needs all solutions. Therefore the use of algebraic constraint equations is proposed as to be able to use strong methods and algorithms from algebraic geometry or numerical algebraic geometry (see Sommese and Wampler (2005)). An important task is to find the simplest algebraic constraint equations, that describe the chains, because after establishing these equations they have to be processed with different solution algorithms. Geometric and algebraic preprocessing is needed before elimination, Groebner base computation or numerical solution processes start. An important feature of algebraic constraint equations is that they provide a complete description of the overall motion behavior of the mechanism. This chapter is therefore a contribution to **Global Kinematics** .

2 Parametrization of $SE(3)$

Euclidean three space is the three dimensional real vector space $\mathbb{R}^3$ together with the usual scalar product $\mathbf{x}^T\mathbf{y} = \sum_{i=1}^{3} x_i y_i$. A Euclidean displacement is a mapping

$$\gamma\colon \mathbb{R}^3 \to \mathbb{R}^3, \quad \mathbf{x} \mapsto \mathbf{M}_R\mathbf{x} + \mathbf{a} \tag{1}$$

where $\mathbf{M}_R \in \mathrm{SO}(3)$ is a proper orthogonal three by three matrix and $\mathbf{a} \in \mathbb{R}^3$ is a vector. The entries of $\mathbf{M}_R$ fulfill the well-known orthogonality condition $\mathbf{M}_R^T \cdot \mathbf{M}_R = \mathbf{I}_3$, where $\mathbf{I}_3$ is the three by three identity matrix.

The group of all Euclidean displacements is denoted by $\mathrm{SE}(3)$. It is a convenient convention to write Eq. (1) as product of a four by four matrix and a four dimensional vector according to[1]

$$\begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \mapsto \mathbf{M} \cdot \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{o}^T \\ \mathbf{a} & \mathbf{M}_R \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}. \tag{2}$$

[1] Note that homogeneous coordinates in this chapter are written in the European notation, with homogenizing coordinate on first place.

The displacements Eq.(2) can be parameterized in different ways. In many applications Euler angles or Rodrigues parameters are used to parameterize the matrix $\mathbf{M}_R$. In the aim of using algebraic tools an algebraic version of the complete 4×4 displacement operator is adopted. This parametrization of the spatial Euclidean transformation matrix $\mathbf{M}\in SE(3)$ is named after *E. Study* (Study (1903)) and can be written as follows (for detailed information on this approach see Husty and Schröcker (2012)):

$$\mathbf{M}=\begin{pmatrix} x_0^2+x_1^2+x_2^2+x_3^2 & 0^\top \\ \mathbf{a} & \mathbf{M}_R \end{pmatrix},\quad \mathbf{a}=\begin{pmatrix} 2(-x_0y_1+x_1y_0-x_2y_3+x_3y_2) \\ 2(-x_0y_2+x_1y_3+x_2y_0-x_3y_1) \\ 2(-x_0y_3-x_1y_2+x_2y_1+x_3y_0) \end{pmatrix} \tag{3}$$

$$\mathbf{M}_R=\begin{pmatrix} x_0^2+x_1^2-x_2^2-x_3^2 & 2(x_1x_2-x_0x_3) & 2(x_1x_3+x_0x_2) \\ 2(x_1x_2+x_0x_3) & x_0^2-x_1^2+x_2^2-x_3^2 & 2(x_2x_3-x_0x_1) \\ 2(x_1x_3-x_0x_2) & 2(x_2x_3+x_0x_1) & x_0^2-x_1^2-x_2^2+x_3^2 \end{pmatrix}$$

The vector $\mathbf{a}$ represents the translational part and $\mathbf{M}_R$ represents the rotational part of the transformation operator $\mathbf{M}$. The parameters $x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3$ which appear in the matrix $\mathbf{M}$ are called *Study*-parameters of the transformation $\mathbf{M}$ and we can consider them as a projective point in $\mathbb{P}^7$. The mapping

$$\begin{gathered} \kappa\colon SE(3)\to P\in\mathbb{P}^7 \\ \mathbf{M}(x_i,y_i)\mapsto (x_0:x_1:x_2:x_3:y_0:y_1:y_2:y_3)^T\neq(0:0:0:0:0:0:0:0)^T \end{gathered} \tag{4}$$

is called *kinematic mapping* and maps each Euclidean displacement of $SE(3)$ to a point P on a quadric $S_6^2\subset\mathbb{P}^7$. In this way, every projective point $(x_0:x_1:x_2:x_3:y_0:y_1:y_2:y_3)\in\mathbb{P}^7$ represents a spatial Euclidean transformation, if it fulfills the following equation of S_6^2:

$$x_0y_0+x_1y_1+x_2y_2+x_3y_3=0 \tag{5}$$

and the inequality:

$$x_0^2+x_1^2+x_2^2+x_3^2\neq 0. \tag{6}$$

Eq. (6) is a normalization term and points with $x_0=x_1=x_2=x_3=0$ do not represent Euclidean transformations, they form the 3-dimensional *exceptional generator* contained in S_6^2. The points on S_6^2 are called *kinematic image points* of the corresponding displacement and the seven dimensional projective space is called *kinematic image space*. More information on this parametrization can be found in Husty and Schröcker (2012), where especially the relation of the Study parameters to dual quaternions is highlighted.

Very often the practical question arises how to compute the Study parameters when an other parameterization is given. Let therefore $\mathbf{A}$ be an orthogonal 3×3

matrix and $\mathbf{a} = (a_1, a_2, a_3)^T$ a 3-dim vector that determine a displacement Eq.(2). Then the entries of Matrix $\mathbf{M}_R$ can be found by one of the four relations:

$$\begin{aligned} x_0 : x_1 : x_2 : x_3 &= 1 + a_{11} + a_{22} + a_{33} : a_{32} - a_{23} : a_{13} - a_{31} : a_{21} - a_{12} \\ &= a_{32} - a_{23} : 1 + a_{11} - a_{22} - a_{33} : a_{12} + a_{21} : a_{31} + a_{13} \\ &= a_{13} - a_{31} : a_{12} + a_{21} : 1 - a_{11} + a_{22} - a_{33} : a_{23} + a_{32} \\ &= a_{21} - a_{12} : a_{31} + a_{13} : a_{23} - a_{32} : 1 - a_{11} - a_{22} + a_{33} \end{aligned} \tag{7}$$

In general, all four proportions of Eq. (7) yield the same result. If, however, $1 + a_{11} + a_{22} + a_{33} = 0$ the first proportion yields $0 : 0 : 0 : 0$ and is invalid. We can use the second proportion instead as long as $a_{22} + a_{33}$ is different from zero. If this happens we can use the third proportion unless $a_{11} + a_{33} = 0$. In this last case we resort to the last proportion which yields $0 : 0 : 0 : 1$. Having computed the first four Study parameters the remaining four parameters $y_0 : y_1 : y_2 : y_3$ can be computed from

$$\begin{aligned} 2y_0 &= \quad a_1x_1 + a_2x_2 + a_3x_3, \\ 2y_1 &= -a_1x_0 + a_3x_2 - a_2x_3, \\ 2y_2 &= -a_2x_0 - a_3x_1 + a_1x_3, \\ 2y_3 &= -a_3x_0 + a_2x_1 - a_1x_2. \end{aligned} \tag{8}$$

Example 1. *A rotation about the x-axis with angle φ is described by the matrix*

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\varphi & -\sin\varphi \\ 0 & 0 & \sin\varphi & \cos\varphi \end{bmatrix}. \tag{9}$$

Its kinematic image, computed via (7) *and* (8) *is*

$$\mathbf{r} = [1 + \cos\varphi : \sin\varphi : 0 : 0 : 0 : 0 : 0 : 0]. \tag{10}$$

As φ varies in $[0, 2\pi)$, $\mathbf{r}$ describes a straight line on the Study quadric which reads after algebraization with half-tangent substitution

$$\mathbf{r}_x = [1 : u : 0 : 0 : 0 : 0 : 0 : 0]. \tag{11}$$

It is easy to see that the other two elementary rotations about $y-$ and $z-$axis can be written in Study parameters as:

$$\mathbf{r}_y = [1 : 0 : v : 0 : 0 : 0 : 0 : 0], \mathbf{r}_z = [1 : 0 : 0 : w : 0 : 0 : 0 : 0]. \tag{12}$$

A second example, which is already a little bit more sophisticated shows the Study representation (or the kinematic image) of a simple $2-R$ chain.

Example 2. *Let* $\mathbf{M}$ *and* $\mathbf{K}$ *be the elementary rotation about the* $x-$ *and the* $y-$*axis. Let furthermore* $\mathbf{N}$ *be a translation along the* $z-$ *axis. The concatenation of these three transformations yields a* $2R$ *chain and its representation is given by* $\mathbf{L} = \mathbf{M}\cdot\mathbf{N}\cdot\mathbf{K}$*;*

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(t) & -\sin(t) \\ 0 & 0 & \sin(t) & \cos(t) \end{bmatrix}, \qquad \mathbf{N} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2a & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(s) & 0 & -\sin(s) \\ 0 & 0 & 1 & 0 \\ 0 & \cos(s) & 0 & \sin(s) \end{bmatrix},$$

$$\mathbf{L} = \mathbf{M}\cdot\mathbf{N}\cdot\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(s) & 0 & -\sin(s) \\ -2\sin(t)a & -\sin(t)\cos(s) & \cos(t) & -\sin(t)\sin(s) \\ 2\cos(t)a & \cos(t)\cos(s) & \sin(t) & \cos(t)\sin(s) \end{bmatrix}.$$

Its kinematic image, computed via (7) *and* (8) *is*

$$\mathbf{l} = \begin{bmatrix} 1+\cos(s)+\cos(t)+\cos(t)\sin(s) \\ \sin(t)+\sin(t)\sin(s) \\ -\sin(s)-\cos(t)\cos(s) \\ -\sin(t)\cos(s) \\ -\sin(t)a(-\sin(s)-\cos(t)\cos(s))-\cos(t)a\sin(t)\cos(s) \\ \cos(t)a(-\sin(s)-\cos(t)\cos(s))-(\sin(t))^2 a\cos(s) \\ \sin(t)a(1+\cos(s)+\cos(t)+\cos(t)\sin(s))-\cos(t)a(\sin(t)+\sin(t)\sin(s)) \\ -\cos(t)a(1+\cos(s)+\cos(t)+\cos(t)\sin(s))-\sin(t)a(\sin(t)+\sin(t)\sin(s)) \end{bmatrix}. \tag{13}$$

It seems that this is much more complicated than the matrix $\mathbf{L}$ *but after algebraization with half-tangent substitution this yields:*

$$\mathbf{l} = [1 : u : v : uv : -uav : av : ua : -a]. \tag{14}$$

Eq.(14) *is the parametric representation of the* $2R$ *chain in Study coordinates.*

Remark 1. *In the Study representation of $SE(3)$ planar and spherical displacements are contained via setting $x_1 = x_2 = y_0 = y_3 = 0$ for planar displacements and $y_0 = y_1 = y_2 = y_3 = 0$ for spherical displacements. Pure translations are obtained in setting $x_0 = 1,\ x_1 = 0,\ x_2 = 0,\ x_3 = 0$.*

3 Constraint equations

In the kinematic analysis or synthesis of a mechanism it is crucial to find a mathematical description of the overall motion capability. In this section we will show different possibilities how a complete set of equations can be found. In this chapter we are interested to find algebraic constraint equations. This restricts first of all the design possibilities of the mechanism, as helical joints can not be described by algebraic (polynomial) equations. As we have seen already in Example 2 the parametric description of a serial chain is straightforward. The question is now, how can we find the corresponding algebraic equations? Generally there are three methods to derive these algebraic constraint equations:

- Using geometric properties of the mechanism. These properties can be for example: one point of the moving system (end effector system) is bound to move on a line, a circle, a sphere or a plane.
- Elimination method: Given the parametric description of the motion of the moving system of the mechanism one can use resultant methods or dialytic elimination methods to derive the algebraic equations. The disadvantage of these methods is that they introduce "spurious" solutions that come from multiple projections onto coordinate spaces. In simple cases this method can be very efficient.
- Linear implicitization algorithm. This algorithm was introduced in Walter and Husty (2010) and guarantees a complete solution of the elimination. The algorithm essentially solves an overconstrained linear system which can be very large in case of high degree polynomial constraint equations.

It is of utmost importance to start any solution process with the simplest set of algebraic equations. There exists always a best adapted coordinate system for a mechanism or at least for one kinematic chain in a more complicated mechanism. When a kinematic chain is represented in its "best" adapted coordinate system then it is called *canonical chain*. Furthermore geometric preprocessing, whenever possible, is highly recommended. Both issues will be explained in detail in the next subsections where we will discuss all three methods with representative examples.

3.1 Geometric constraint equations

The use of geometric properties to derive constraint equations will be demonstrated with a planar 3-RRR parallel manipulator. This manipulator has been dis-

cussed by many authors. Many details can be found in Bonev (2002).

The manipulator consists of three anchor points P_1, P_2, P_3 in the base and three points p_1, p_2, p_3 in a moving system (Fig.1). The three pairs of points are connected via three planar dyads. The links in the dyads are denoted by $l_i, k_i, i = 1, \ldots 3$. The revolute joints at the base points P_1, P_2, P_3 are active. All other revolute joints are passive. For the mathematical description we use the planar dis-

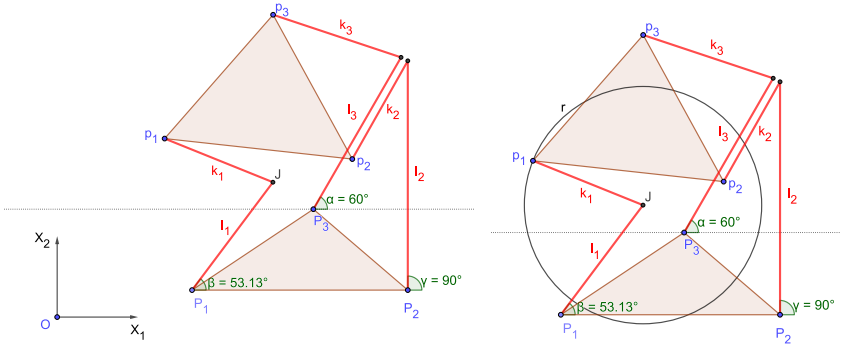


Figure 1. 3-RRR-parallel manipulator

placement subgroup parameterized by Study parameters:

$$\mathbf{M} = \begin{bmatrix} {x_0}^2 + {x_3}^2 & 0 & 0 & 0 \\ -2x_0y_1 + 2x_3y_2 & {x_0}^2 - {x_3}^2 & -2x_0x_3 & 0 \\ -2x_0y_2 - 2x_3y_1 & 2x_0x_3 & {x_0}^2 - {x_3}^2 & 0 \\ 0 & 0 & 0 & {x_0}^2 + {x_3}^2 \end{bmatrix}. \tag{15}$$

Without loss of generality we can cancel the last row and the last column and obtain the 3×3 planar transformation matrix

$$\mathbf{A} = \begin{bmatrix} {x_0}^2 + {x_3}^2 & 0 & 0 \\ -2x_0y_1 + 2x_3y_2 & {x_0}^2 - {x_3}^2 & -2x_0x_3 \\ -2x_0y_2 - 2x_3y_1 & 2x_0x_3 & {x_0}^2 - {x_3}^2 \end{bmatrix}. \tag{16}$$

From Fig.1, right it is obvious that, in case of fixed input parameters α, β, γ, the platform points p_1, p_2, p_3 are bound to be on three circles having the "knee joints" (e.g. J in Fig.1) as centers and radii k_i. Using homogeneous coordinates $(X_0, X_1, X_2)^T$ in the base coordinate system a circle with center $(1, m, n)^T$ can be written

$$X_1^2 + X_2^2 - 2mX_0X_1 - 2nX_0X_2 + (m^2 + n^2 - r^2)X_0^2 = 0 \tag{17}$$

Let $\mathbf{p} = (1, x, y)^T$ be the coordinates of a point in the moving system, then its coordinates in the fixed base system are given by

$$\mathbf{P} = \mathbf{A}\mathbf{p} = \begin{bmatrix} x_0^2 + x_3^2 \\ -2x_0y_1 + 2x_3y_2 + (x_0^2 - x_3^2)x - 2x_0x_3y \\ -2x_0y_2 - 2x_3y_1 + 2x_0x_3x + (x_0^2 - x_3^2)y \end{bmatrix}. \tag{18}$$

Substituting Eq.(18) into the circle equation Eq.(17) yields after factoring and cancelling $x_0^2 + x_3^2$ the general circle constraint equation

$$\begin{aligned}&(x^2 + y^2 + m^2 - 2mx + n^2 - 2ny - r^2)x_0^2 + 4(my - nx)x_0x_3 + 4(m - x)x_0y_1 + \\ &4(n - y)x_0y_2 + (x^2 + y^2 + m^2 + 2mx + n^2 + 2ny - r^2)x_3^2 + 4(y + n)x_3y_1 - \\ &4(x + m)x_3y_2 + 4y_1^2 + 4y_2^2 = 0.\end{aligned} \tag{19}$$

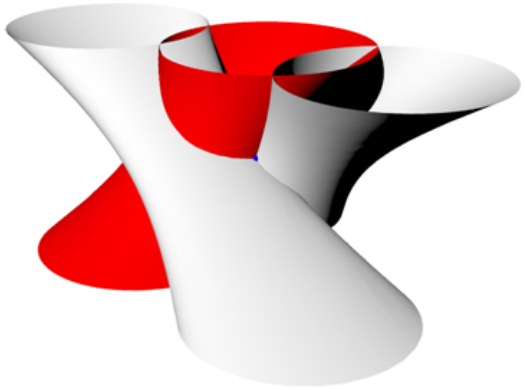


Figure 2. Constraint hyperboloids in kinematic image space

Note that in Eq.(19) all expressions in round brackets are design constants. Therefore the circle constraint equation allows a simple geometric interpretation in the three dimensional kinematic image space of planar kinematics: it represents a hyperboloid which passes, independently of the choice of the design constants, through the points $J_1(0, 1, I)^T$ and $J_2(0, 1, -I)^T$ (see Bottema and Roth (1979), pp.409). A further observation is that the expression $4y_1^2 + 4y_2^2$ in Eq.(19) does not contain any design variable.

An important next step is to use the best adapted coordinate system. Without loss of generality P_1, p_1 can be the origins of fixed and moving coordinate system and P_2, p_2 can be on the X_1 resp. x-axis of both systems. Therefore we have the

following assignments

$$P_1 = [1,0,0]^T, \quad P_2 = [1,A_2,0]^T, \quad P_3 = [1,A_3,B_3]^T,$$
$$p_1 = [1,0,0]^T, \quad p_2 = [1,a_2,0]^T, \quad p_3 = [1,a_3,b_3]^T.$$

In a next step one can observe that the knee joints also move on circles. These circles can be parameterized as follows

$$m_1 = l_1 \tfrac{1-u^2}{1+u^2}, \quad m_2 = l_2 \tfrac{1-v^2}{1+v^2} + A_2, \quad m_3 = l_3 \tfrac{1-w^2}{1+w^2} + A_3,$$
$$n_1 = l_1 \tfrac{2u}{1+u^2}, \quad n_2 = l_2 \tfrac{2v}{1+v^2}, \qquad\quad n_3 = l_3 \tfrac{2w}{1+w^2} + B_3.$$

Substituting all assignments into the general circle constraint equations using $r = k_i, i = 1\ldots3$ for the respective legs, we obtain three constraint equations which completely describe the 3-RRR parallel mechanism:

$$\begin{aligned}
h_1 :\ & (l_1^2 - k_1^2)(x_0^2 + x_3^2) + 4l_1 \left(\frac{1-u^2}{1+u^2}(x_0 y_1 - x_3 y_2) + \frac{2u}{1+u^2}(x_0 y_2 + x_3 y_1) \right) + \\
& 4(y_1^2 + y_2^2) = 0, \\
h_2 :\ & \left(\frac{r_1 r_2 v^2 + r_3 r_4)}{v^2+1} \right) x_0^2 + \left(\frac{r_5 r_6 v^2 + r_7 r_8)}{v^2+1} \right) x_3^2 - 4a_2(x_0 y_1 + x_3 y_2) + \\
& 4(l_2 \frac{1-v^2}{1+v^2} + A_2)(x_0 y_1 - x_3 y_2) + 4l_2 \frac{2v}{1+v^2}(a_2 x_0 x_3 + x_0 y_2 + x_3 y_2) + 4(y_1^2 + y_2^2) = 0, \\
& r_1 := A_2 - k_2 - a_2 - l_2,\ r_2 := A_2 + k_2 - a_2 - l_2,\ r_3 := A_2 - k_2 - a_2 + l_2, \\
& r_4 := A_2 + k_2 - a_2 + l_2,\ r_5 := A_2 + k_2 + a_2 - l_2,\ r_6 := A_2 - k_2 + a_2 - l_2, \\
& r_7 := A_2 + k_2 + a_2 + l_2,\ r_8 := A_2 - k_2 + a_2 + l_2, \qquad (20) \\
h_3 :\ & \frac{(q_1^2 + q_2)w^2 + 4l_3(B_3 - b_3)w + q_4^2 + q_2 q_3}{1+w^2} x_0^2 + \\
& \left(4 \left(\frac{l_3(1-w^2)}{w^2+1} + A_3 \right) b_3 - (4(\frac{2wl_3}{1+w^2} + B_3))a_3) \right) x_0 x_3 \\
& \left(-4a_3 + 4l_3 \frac{1-w^2}{w^2+1} + 4A_3 \right) x_0 y_1 + \left(-4b_3 + \frac{8wl_3}{(w^2+1} + 4B_3 \right) x_0 y_2 + \\
& \left(4b_3 + \frac{8wl_3}{w^2+1} + 4B_3 \right) x_3 y_1 + \left(-4a_3 - 4l_3 \frac{1-w^2}{w^2+1} - 4A_3 \right) x_3 y_2 \\
& \frac{(q_5^2 + q_6 q_7)w^2 + 4l_3(B_3 b_3)w + q_8^2 + q_6 q_7}{1+w^2} x_3^2 + 4(y_1^2 + y_2^2) = 0, \\
& q_1 := A_3 - a_3 - l_3,\ q_2 := -b_3 + B_3 - k_3,\ q_3 := -b_3 + B_3 + k_3,\ q_4 := A_3 - a_3 + b_3, \\
& q_5 := A_3 - l_3 + a_3,\ q_6 := b_3 + B_3 + k_3,\ q_7 := b_3 + B_3 - k_3,\ q_8 := A_3 + a_3 + l_3.
\end{aligned}$$

Using the three equations h_1, h_2, h_3 and a normalization condition one can solve the direct kinematics (DK), the inverse kinematics (IK), the forward and the inverse

singularities completely. These are just the most important kinematic tasks which will be shown in the sequel, but the equations can be used also for other tasks.

Direct Kinematics In the direct kinematics all design variables and the three input parameters are given and the pose of the moving platform has to be computed. Although the univariate polynomial of the direct kinematics of this manipulator can be computed without assigning values to the design and input variables we will show the algorithm for the example that was already used to produce Fig.1. Before values assigned to the design parameters one should have a closer look at the constraint equations. It is important to know that several manipulations are allowed, such as addition, subtraction, multiplication with arbitrary polynomials and constants without changing the solution set of the constraint equations. A stringent mathematical formulation of this statement will be given in Section 4. In case of the quadratic constraint equations h_1, h_2, h_3 one can observe that a subtraction $h_{1j} := h_j - h_1, j = 2,3$ creates two new equations that are linear in the variables y_1, y_2. It would be tempting to create one more difference equation, but it is easy to show that a third difference equation would be redundant and would not contain any new information. At least one of the three original equations has to be kept. The detailed solution process will be shown in the following example.

Example 3. *The following design variables are assigned to a 3-RRR planar parallel manipulator:*

$$A_2 = 16, A_3 = 9, B_3 = 6, a_2 = 14, a_3 = 7, b_3 = 10, l_1 = 10, l_2 = 17, l_3 = 13,$$
$$k_1 = \sqrt{75}, k_2 = \sqrt{70}, k_3 = 10.$$

Furthermore the three input variables are given by

$$u = \frac{1}{2}, v = 1, w = \frac{\sqrt{3}}{3}.$$

The three constraint equations simplify considerably

$$\begin{aligned}
h_1 &: \ 25x_3^2 + 32x_3y_1 - 24x_3y_2 + 4y_1^2 + 4y_2^2 + 24y_1 + 32y_2 + 25 = 0, \\
h_2 &: \ 1119x_3^2 + 68x_3y_1 - 120x_3y_2 + 4y_1^2 + 4y_2^2 - 952x_3 + 8y_1 + 68y_2 + 223 = 0, \qquad (21) \\
h_3 &: \ 620x_3 + \frac{2025{x_3}^2}{4} - 130\sqrt{3} - \frac{191}{4} + 40y_1x_3 + 34y_1 - 90x_3y_2 - 40y_2 + 4{y_1}^2 + \\
&\quad 4{y_2}^2 + \left(20{x_3}^2 + 4y_1x_3 - 28x_3 + 4y_2\right)\left(\frac{13\sqrt{3}}{2} + 6\right) + \left({x_3}^2 + 1\right)\left(\frac{13\sqrt{3}}{2} + 6\right)^2 = 0.
\end{aligned}$$

In this set of equation $x_0 = 1$ has been used as normalization. This normalization excludes all 180° turns. For a complete analysis one has to check if $x_0 = 0$ is

a solution. In this example it is easy to see that this is not the case. First the equations h_{12} and h_{13} are solved for y_1 and y_2 and the result is substituted into h_1. This yields the univariate polynomial:

$$\begin{aligned}&1012018158645001\,{x_3}^6+373126531431576\,\sqrt{3}{x_3}^5+828170897821956\,\sqrt{3}{x_3}^4\\&-1870238901095276\,{x_3}^5-3830372502668712\,\sqrt{3}{x_3}^3-309592552617273\,{x_3}^4-\\&1367698801300104\,\sqrt{3}{x_3}^2+5703740216839288\,{x_3}^3+2552443644341760\,\sqrt{3}x_3+\\&2666944473586507\,{x_3}^2-584052482710476\,\sqrt{3}-4438269370622172\,x_3+\\&1009620776386125=0.\end{aligned}\tag{22}$$

This polynomial has four real and two complex roots for x_3, which is the rotation angle of the moving system :

$$\begin{aligned}&-0.01155649481-.8571792684\,I,\ \ -0.01155649481+.85717926841\,I,\\&-0.05446878513,\ \ .17472650281,\ \ .3874512485,\ \ .7248336963.\end{aligned}\tag{23}$$

This six values have to be substituted into the solution for y_1 and y_2, which then yields the pose variables for the six solutions of the DK of this manipulator. This procedure is shown for the first real solution. Substituting this solution into the solutions for y_1 and y_2 yields

$$y_1=1.3224316875,\qquad y_2=-5.5627826306.\tag{24}$$

Now all pose parameters are determined and can be substituted into the transformation matrix Eq.(18)*:*

$$\mathbf{A}=\begin{bmatrix}1.002966848554 & 0 & 0\\ -2.038867351357 & 0.997033151446 & 0.1089375702689\\ 11.26962775615 & -0.10893757026889 & 0.99703315144598\end{bmatrix}.$$

Note that this matrix is not normalized (has no 1 at the leftmost entry of the first row). This is due to the fact that in the image space the normalization $x_0=1$ was used. By dividing the matrix with this entry one obtains the normalized matrix

$$\mathbf{A}=\begin{bmatrix}1 & 0.0 & 0.0\\ -2.032836234114 & 0.9940838551976 & 0.1086153250488\\ 11.23629138131 & -0.1086153250488 & 0.9940838551976\end{bmatrix}.$$

Multiplying $\mathbf{A}$ with the coordinates of the points p_1,p_2,p_3 one obtains their coordinates in the base system and the pose of the end-effector is given as solution of

the DK problem.

$$p_1^b = [1, -2.032836234114, 11.2362913813]^T,$$
$$p_2^b = [1, 11.8843377387, 9.715676830627]^T,$$
$$p_3^b = [1, 6.01190400276, 20.41682265796]^T.$$

Fig.3 shows this solution.

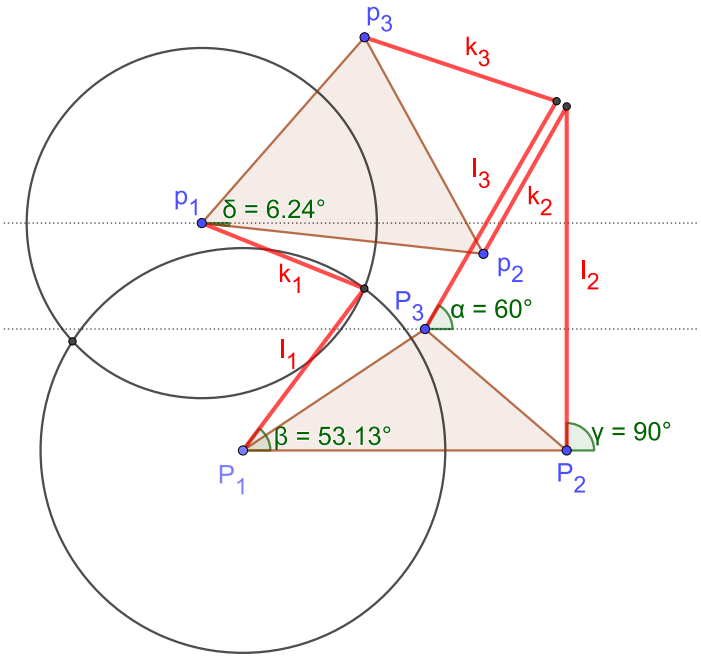


Figure 3. DK solution 3-RPR

The rotation angle of the platform system can be calculated via the inverse of the half tangent substitution

$$\cos\varphi = \frac{1-x_3^2}{1+x_3^2} \wedge \sin\varphi = \frac{2x_3}{1+x_3^2} \wedge x_3 = -0.05446878513$$
$$\Rightarrow \varphi = -6.23550128120984°.$$

Note that the input angles were

$$u = \frac{1}{2} \equiv 53.13010235°,\ v = 1 \equiv 90°,\ w = \frac{\sqrt{3}}{3} \equiv 60°.$$

Inverse Kinematics In the inverse kinematics (IK) the design variables and the pose parameters are given. The same constraint equations can be used. We continue with Example 3.

Example 4 (continuation of Example 3)**.** *Substituting the design variables into Eq.*(20) *yields*

$$
\begin{aligned}
h_1 &: 25u^2x_3^2+40u^2x_3y_2+4u^2y_1^2+4u^2y_2^2-40u^2y_1+80ux_3y_1+25u^2+80uy_2+25x_3^2-\\
&\quad 40x_3y_2+4y_1^2+4y_2^2+40y_1+25=0\\
h_2 &: 99v^2x_3^2-52v^2x_3y_2+4v^2y_1^2+4v^2y_2^2-60v^2y_1+136vx_3y_1+155v^2-1904vx_3+\\
&\quad 136vy_2+2139x_3^2-188x_3y_2+4y_1^2+4y_2^2+76y_1+291=0 \qquad (25)\\
h_3 &: 165w^2x_3^2+64w^2x_3y_1-12w^2x_3y_2+4w^2y_1^2+4w^2y_2^2-328w^2x_3-44w^2y_1-16w^2y_2+\\
&\quad 832wx_3^2+104wx_3y_1+37w^2-728wx_3+104wy_2+997x_3^2+64x_3y_1-116x_3y_2+4y_1^2+\\
&\quad 4y_2^2-208w+712x_3+60y_1-16y_2+14=0.
\end{aligned}
$$

This system of equations in the unknowns u,v,w allows eight different solutions when a set of pose parameters are given. We use the results of Eqns.(22) *and* (23) *to compute the inverse kinematics of the pose with pose parameters*

$$x_0=1, \quad x_3=-0.05446878513, \quad y_1=1.3224316875, \quad y_2=-5.5627826306.$$

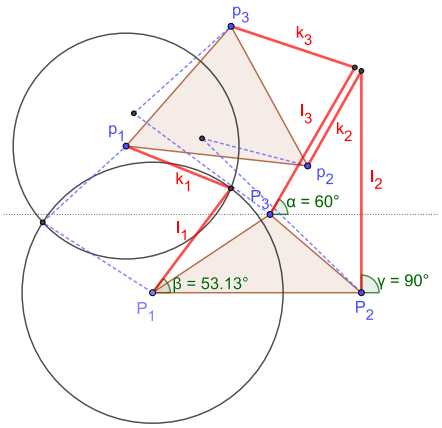


Figure 4. IK solution of 3-RRR manipulator

The solutions for the resulting set of equations are

$$u = .500000, v = 1.00000, w = .57735, \;\; u = .500000, v = 1.00000, w = 3.0254,$$
$$u = 3.41747, v = 1.00000, w = .57735, \;\; u = 3.41747, v = 1.00000, w = 3.0254,$$
$$u = .500000, v = 2.46987, w = .57735, \;\; u = .500000, v = 2.46987, w = 3.0254,$$
$$u = 3.41747, v = 2.46987, w = .57735, \;\; u = 3.41747, v = 2.46987, w = 3.0254.$$

Fig.4 shows the complete set of eight possible solutions, consisting of all possible combinations of the pairs of solutions of each equation $h_i, i = 1,2,3$. The second solutions for each dyad are drawn in dashed lines. Each solution yields a working mode of the manipulator. The first set of solutions (the first working mode) is the one we have started with in the DK.

Singularities The system of three constraint equations $\mathscr{S} = \{h_1, h_2, h_3, n\}$ augmented with the normalization condition $n : x_0^2 + x_3^2 = 1$ (Eq.(20)) comprises the forward and the inverse map. Therefore it can be used to compute the singularities of both maps. Referring to Gosselin and Angeles (1990) the singularities of both maps are given by

$$\mathbf{J}_o \dot{\mathbf{y}} + \mathbf{J}_i \dot{\mathbf{t}} = 0, \tag{26}$$

where

$$\mathbf{J}_o = \begin{bmatrix} \frac{\partial n}{\partial x_0} & \frac{\partial n}{\partial x_3} & 0 & 0 \\ \frac{\partial h_1}{\partial x_0} & \frac{\partial h_1}{\partial x_3} & \frac{\partial h_1}{\partial y_1} & \frac{\partial h_1}{\partial y_2} \\ \frac{\partial h_2}{\partial x_0} & \frac{\partial h_2}{\partial x_3} & \frac{\partial h_2}{\partial y_1} & \frac{\partial h_2}{\partial y_2} \\ \frac{\partial h_3}{\partial x_0} & \frac{\partial h_3}{\partial x_3} & \frac{\partial h_3}{\partial y_1} & \frac{\partial h_3}{\partial y_2} \end{bmatrix}, \quad \mathbf{J}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{\partial h_1}{\partial u} & 0 & 0 \\ 0 & 0 & \frac{\partial h_2}{\partial v} & 0 \\ 0 & 0 & 0 & \frac{\partial h_3}{\partial w} \end{bmatrix},$$

$$\dot{\mathbf{y}} = [\dot{x}_0, \dot{x}_3, \dot{y}_1, \dot{y}_2]^T, \quad \dot{\mathbf{t}} = [0, \dot{u}, \dot{v}, \dot{w}]^T.$$

$\mathbf{J}_o$ is the Jacobian matrix of the forward map and $\mathbf{J}_i$ is the Jacobian matrix of the inverse map of the manipulator. In the literature the singularities of the inverse map are sometimes called type 1 singularities and those of the forward map are called type 2 singularities Bonev (2002) [2]. The redundant motion which exists either on the platform or in the joints can be infinitesimal or finite. The infinitesimal case is treated in the following sections. The finite case, which occurs only at the presence of specific design or pose parameters, can be detected in the solution of the direct or inverse kinematics described in the previous sections. In the solution process

[2] A refined classification, taking into account special situations which can occur for special design parameters of the manipulator was given in Zlatanov et al. (1994).

it happens that the elimination stops before a univariate polynomial can be found. Then the manipulator allows a self-motion, i.e. the platform or an input chain will move without active input. An example of this phenomenon will be shown later in Section 4 on page 55.

Forward singularities From kinematic point of view a forward singularity occurs when a parallel manipulator allows an infinitesimal (local) mobility when all the inputs are locked. Mathematically this corresponds to the condition $\dot{\mathbf{t}} = [0,0,0,0]^T$ and substitution of this condition into Eq.(26) yields

$$\mathbf{J}_o \dot{\mathbf{y}} = 0. \tag{27}$$

Eq.(27) represents a homogeneous system of linear equations. From linear algebra it is well know that this system allows a non trivial solution if and only if $\det \mathbf{J}_o = 0$ holds. Computing this determinant without assigning design variables is feasible but without practical use. One has to realize that this determinant still contains all pose parameters x_0, x_3, y_1, y_2 and all input parameters u, v, w. Adding $\det \mathbf{J}_o = 0$ to the system of constraint equations $\mathscr{S}$ yields a system of five equations. In this system one can for example eliminate the input parameters u, v, w to obtain an equation in the pose parameters x_0, x_3, y_1, y_2 and a normalization condition that represent in the kinematic image space all poses which are forward singular. The disadvantage of this equation is that the working modes cannot be distinguished. Elimination of the pose parameters yield the singular set in the joint space. The practicality of the latter is although questionable, because once this equation is obtained one has to perform the direct kinematics again and only one of the obtained six solutions yields a singular pose!

Example 5 (Continuation of Example 3)**.** *Taking the system of constraint equations* (25) *and computing the determinant of* $\mathbf{J}_o$ *yields a polynomial* h_4 *of degree 10 in the unknowns* $x_0, x_3, y_1, y_2, u, v, w$*. Together with the original constraint equations* h_1, h_2, h_3 *one obtains a system of four algebraic equations that allows to derive a single polynomial in the Study parameters or in the joint parameters which will comprise all forward singular poses of the manipulator. A suitable normalization is still possible. The zero set of this polynomial describes a surface in the kinematic image space or the joint space depending on which of the variables have been eliminated. One can either eliminate the input or the output variables. Due to the above mentioned fact that computing the singularity surface in the joint space needs an additional computation of the forward kinematics, only the elimination of the input parameters will be shown. To obtain a single polynomial we compute the following sequence of resultants of equations:* $h_{14} = res(h_1, h_4, u)$ *is the resultant of* h_1 *and* h_4 *that eliminates u. Then we take* $h_{142} = res(h_{14}, h_2, v)$*, which eliminates v and finally we take* $h_{1423} = res(h_{142}, h_3, w)$ *which eliminates w to arrive*

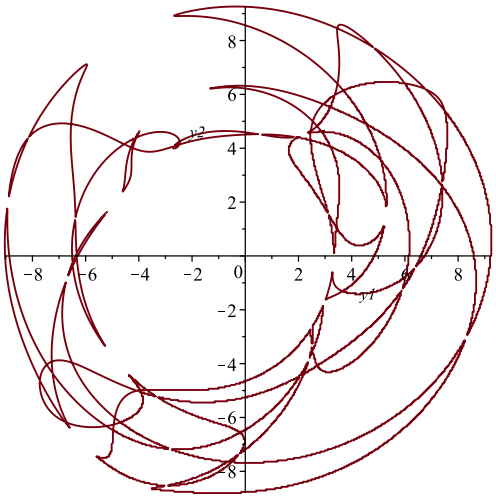


Figure 5. Section of forward singularity surface at $x_0 = 1, x_3 = 0$ in image space

at a polynomial of degree 44 in the remaining pose parameters x_0, x_3, y_1, y_2. *As spurious solutions might occur it is recommended to use a different sequence of elimination and compare the two final polynomials. In case they are different one has to look for their greatest common divisor. In case of the example the second sequence* $h_{34} = res(h_3, h_4, w) \rightarrow h_{342} = res(h_{34}, h_2, v) \rightarrow h_{3421} = res(h_{342}, h_1, u)$ *was used and this elimination yielded the same polynomial of degree 44*[3]. *The degree of the polynomial is so high because it describes all forward singularities of all working modes. In the kinematic image space the polynomial describes a surface of order 44. Fig. 5 shows a section of this surface at* $x_0 = 1, x_3 = 0$. *To show a single singular pose of the manipulator we take the uppermost point of this curve on the* y_2 *axis of Fig.5. Substituting* $x_0 = 1, x_3 = 0, y_1 = 0$ *into the polynomial of degree 44 yields a univariate polynomial which is to big to be displayed completely*

$$
\begin{aligned}
&721130272470513326489 6y_2^{44} - 847414787593148584427 52y_2^{43} - \ldots \\
&- 120507924001447407269584976652803693359375000000000y_2 + \\
&217755257309527371670821082331735733184814453125 00 = 0.
\end{aligned}
$$

This polynomial has twelve real roots as can be seen also in Fig.5:

[3] The degree of this polynomial disagrees with the results found in the thesis Bonev (2002).

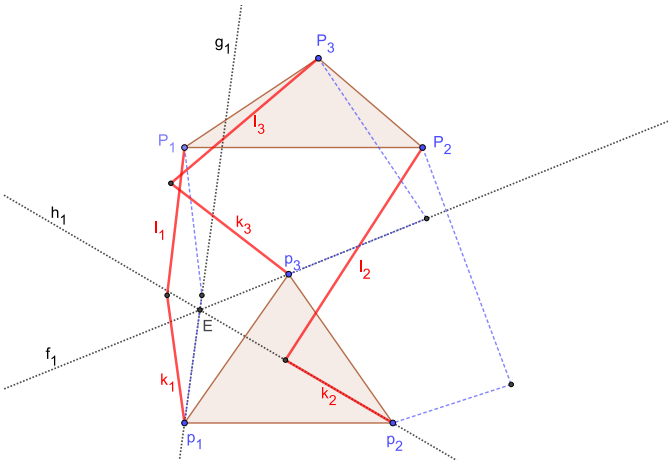


Figure 6. Forward singularity 3-RRR

$$-8.773883747, -7.680052573, -7.189867449, -6.457249958,$$
$$-5.253827948, -4.649535025, 4.507174754, 4.592908938,$$
$$6.218593708, 6.317560304, 8.566805064, 9.256203129.$$

Fig.6 shows the singular pose corresponding to the root $y_2 = 9.256203129$. *The three lines* f_1, g_1, h_1 *covering three end links of the dyads intersect in one point E, which is the instantaneous center of the redundant infinitesimal rotation in this singular pose. The pose belongs to a working mode consisting of one red and two blue dashed input dyads. Note that the moving platform is below the fixed platform in the displayed pose!*

Inverse singularities From kinematic point of view an inverse singularity occurs when the manipulator allows an infinitesimal (local) mobility of an input chain when the platform is not moving. Mathematically this corresponds to the condition $\dot{\mathbf{y}} = [0,0,0,0]^T$ and substitution of this condition into Eq.(26) yields

$$\mathbf{J}_i \dot{\mathbf{t}} = 0. \tag{28}$$

It is quite obvious from Eq.(26) that this determinant factors into three parts:

$$
\begin{aligned}
h_5 : & \big[(B_3x_0^2 + B_3x_3^2 - 2a_3x_0x_3 - b_3x_0^2 + b_3x_3^2 + 2x_0y_2 + 2x_3y_1)w^2 + \\
& (2A_3x_0^2 + 2A_3x_3^2 - 2a_3x_0^2 + 2a_3x_3^2 + 4b_3x_0x_3 + 4x_0y_1 - 4x_3y_2)w \\
& -2x_0y_2 - 2x_3y_1 - B_3x_0^2 - B_3x_3^2 + 2a_3x_0x_3 + b_3x_0^2 - b_3x_3^2\big]\, l_3 \cdot \\
& \big[(-a_2x_0x_3 + x_0y_2 + x_3y_1)v^2 + (A_2x_0^2 + A_2x_3^2 - a_2x_0^2 + a_2x_3^2)v + \\
& a_2x_0x_3 + 2vx_0y_1 - 2vx_3y_2 - x_0y_2 - x_3y_1)l_2\big] \cdot \\
& \big[(u^2x_0y_2 + u^2x_3y_1 + 2ux_0y_1 - 2ux_3y_2 - x_0y_2 - x_3y_1)l_1\big] = 0.
\end{aligned}
\tag{29}
$$

Each factor can be treated separately to determine the input singularities for the corresponding leg. For example when the third factor is used together with equation h_1, the input parameter u can be eliminated to derive all poses of the moving platform in which the first leg is singular. After elimination the following equation is obtained:

$$
\begin{aligned}
& (y_1^2 + y_2^2)(x_0^2 + x_3^2)(k_1^2x_0^2 + k_1^2x_3^2 - 2k_1l_1x_0^2 - 2k_1l_1x_3^2 + l_1^2x_0^2 + l_1^2x_3^2 - 4y_1^2 - 4y_2^2) \cdot \\
& (k_1^2x_0^2 + k_1^2x_3^2 + 2k_1l_1x_0^2 + 2k_1l_1x_3^2 + l_1^2x_0^2 + l_1^2x_3^2 - 4y_1^2 - 4y_2^2) = 0.
\end{aligned}
\tag{30}
$$

The first two factors in this equation have only complex solutions and can be neglected, the third and the fourth factor yield two surfaces in the kinematic image space which turn out to be two hyperboloids (Fig.7). One of the hyperboloids represents all poses where the first leg is totally stretched and the other represents the the poses where the first leg is folded. Additionally these hyperboloids represent poses where the manipulator is on the boundary of its workspace.

Example 6 (Continuation of Example 3)**.** *Substituting the design parameters of Example 3 into Eq.*(30) *yields the equations*

$$
\begin{aligned}
& (16y_1^2\sqrt{3} + 16y_2^2\sqrt{3} + 25x_0^2 + 25x_3^2 - 28y_1^2 - 28y_2^2) = 0 \\
& (16y_1^2\sqrt{3} + 16y_2^2\sqrt{3} - 25x_0^2 - 25x_3^2 + 28y_1^2 + 28y_2^2) = 0.
\end{aligned}
\tag{31}
$$

To derive a single inverse singular pose a point on one of the hyperboloids is chosen by setting $x_0 = 1, y_1 = -1, x_3 = -\frac{1}{4}$. *This yields the equation*

$$16\sqrt{3} + 16y_2^2\sqrt{3} - \frac{23}{16} - 28y_2^2 = 0.$$

One of the solutions of this equations is

$$y_2 = \frac{1}{8}\,\frac{\sqrt{\left(7 - 4\sqrt{3}\right)\left(256\sqrt{3} - 23\right)}}{4\sqrt{3} - 7}.$$

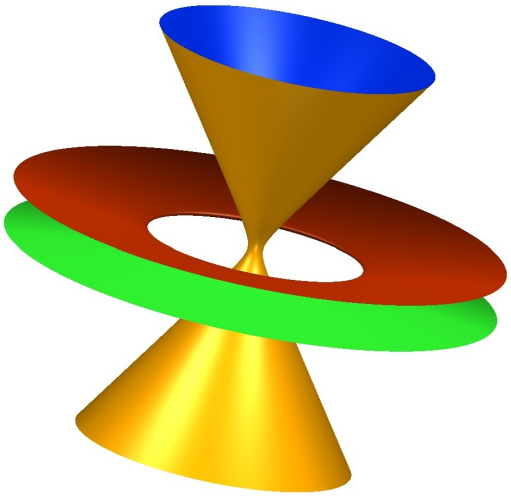


Figure 7. Inverse singularity surfaces in kinematic image space

Substituting this value and the chosen parameters $x_0 = 1, y_1 = -1, x_3 = -\frac{1}{4}$ *into the matrix Eq.*(16) *one obtains the transformation matrix*

$$\mathbf{A} = \begin{bmatrix} \frac{17}{16} & 0 & 0 \\ 2 - \frac{1}{16}\frac{\sqrt{(7-4\sqrt{3})(256\sqrt{3}-23)}}{4\sqrt{3}-7} & \frac{15}{16} & \frac{1}{2} \\ -\frac{1}{4}\frac{\sqrt{(7-4\sqrt{3})(256\sqrt{3}-23)}}{4\sqrt{3}-7} - \frac{1}{2} & -\frac{1}{2} & \frac{15}{16} \end{bmatrix}.$$

This transformation yields the singular pose of Fig.8.

In case of the inverse singularities we will also compute the singularity surface in the joint space, to exemplify the statement made above about this surface. We have to start with the system of equations $\mathscr{S} = \{h_1, h_2, h_3, h_5\}$ consisting of the three constraint equations and the equation of the determinant of the Jacobian $\mathbf{J}_i$. Now the pose parameters x_0, x_3, y_1, y_2 have to be eliminated as to obtain a single equation in the joint parameters u, v, w. To simplify the computation we use the normalization condition $x_0 = 1$ and compute y_1 and y_2 from the difference equations $h_{12} = h_1 - h_2$ and $h_{13} = h_1 - h_3$ and substitute the solutions for y_1 and y_2 into h_1 and the third factor of h_5. This yields two equations in u, v, w and x_3 from which x_3 can be eliminated using resultant methods. The result is an equation of

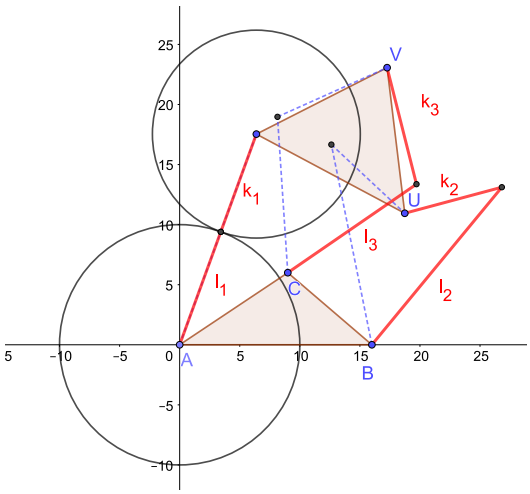


Figure 8. Inverse singularity of a 3-RRR parallel manipulator

degree 28 in the input parameters u, v, w representing the singularity surface $\mathscr{F}$ in joint space, which is much to big to be displayed.

Example 7 (Continuation of Example 3)**.** *By substituting the design parameters of Example 3 into the equation of $\mathscr{F}$ a picture of the singularity surface can be shown (Fig.9).*

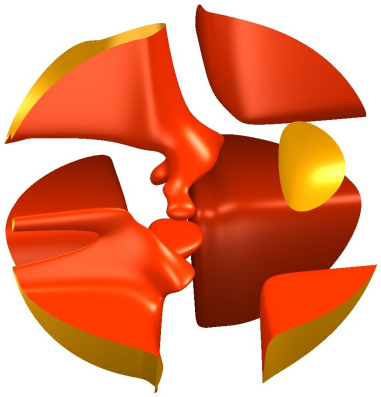


Figure 9. Inverse singularity surface in joint space

Now a single inverse singular pose is specified by setting $v = 1, w = 1$. *Substituting these values into the equation of* $\mathscr{F}$ *yields a polynomial of degree 12:*

$$-818544231103537077\,u^{12} + 3944558869211116800\,u^{11} - 9479437871294979126\,u^{10} + 15757734038472565344\,u^9 - 19686222594094813627\,u^8 + 18937447175451975936\,u^7 - 14224778429264845268\,u^6 + 8254317529677310144\,u^5 - 3535122247251754347\,u^4 + 1029296909802009856\,u^3 - 175954826762074230\,u^2 + 12130149459227104\,u + 220928936206427 = 0,$$

which has six real roots for u

$$-.01476395296624513,\ .2723109282032673,\ .5270689907577652,$$
$$.6265818046074013,\ .6645283032344548,\ 1.402004907351409.$$

We take the root $u = .5270689907577652$ *which determines together with* $v = 1, w = 1$ *a point on* $\mathscr{F}$. *Now the input for the direct kinematics of this manipulator is determined and we can apply the procedure developed above to compute the direct kinematics. The computation yields six real poses, one of them is the singular one! The mapping from joint space to the Cartesian space is 1 to 6! Figs.10-12 show all six solutions. In solution Nr. 2 the first leg is singular. It must be noted that the whole computation was done only for the first leg, because the solutions for* y_1 *and* y_2 *were substituted into* h_1. *One also could have used* h_2 *or* h_3 *at this step. A complete inverse singularity analysis in joint space yields three surfaces, each of them corresponds to the singularities of one leg.*

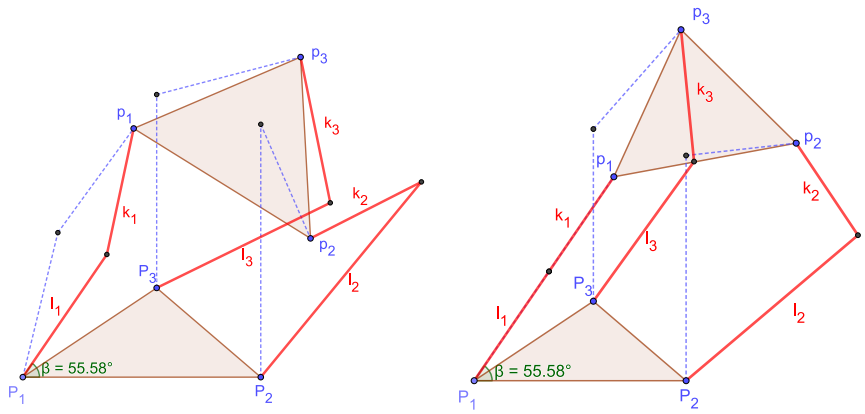
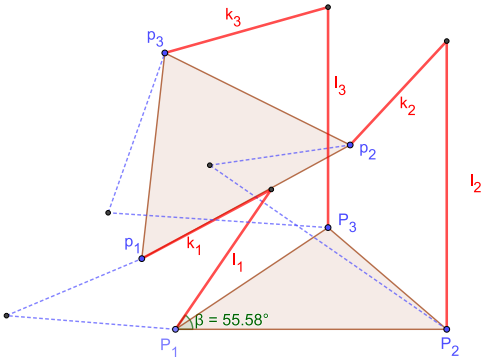


Figure 10. Solution 1 Solution 2



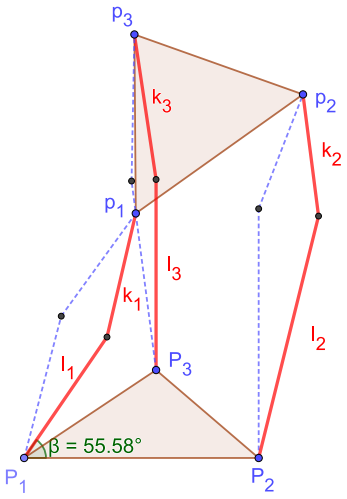
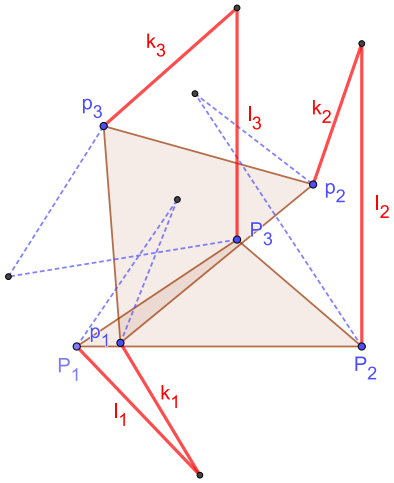


Figure 11. Solution 3 Solution 4



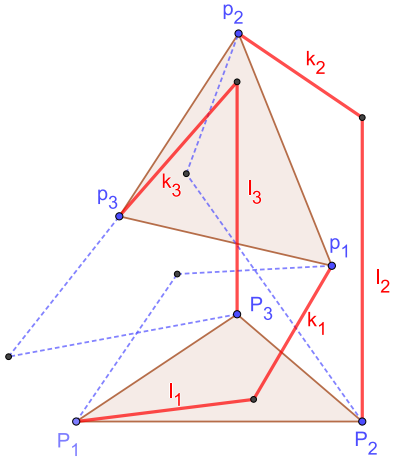


Figure 12. Solution 5 Solution 3

3.2 Elimination Method

This method follows a simple recipe: Write the forward kinematics of the kinematic chain and than eliminate the motion parameters. When n is the degree of

freedom of the kinematic chain then the number m of constraint equations (in general) to be expected is $m = 6 - n$. Because of projection roots, that are introduced in every single elimination step the degree of the resulting constraint equations is generally very high. Because of this fact this method is recommended only for simple chains. An example where this method immediately yields a result are the constraint equations of a spatial 2-R chain. It was already shown in Selig (2005) that a 2-R chain can be represented by four linear equations in Study coordinates. We demonstrate this statement using Example 2.

Example 8. *In Example 2 it was shown that a 2-R chain, having the x-axis and a parallel line to the y-axis as rotation axes has the parametric representation*

$$\mathbf{l} = [1 : u : v : uv : -uav : av : ua : -a].$$

This homogeneous vector equation consists of eight component equations:

$$\rho x_0 = 1, \rho x_1 = u, \rho x_2 = v, \rho x_3 = uv, \rho y_0 = -auv, \rho y_1 = av, \rho y_2 = au, \rho y_3 = -a.$$

The first equation immediately yields $\rho = 1$ and the homogenizing equation $x_0 = 1$. The second and the third equation can be used to eliminate the motion parameters u and v. This yields five equations which seem to be too much because we have to expect four equations:

$$x_3 - x_1 x_2 = 0, \quad y_0 + a x_1 x_2 = 0, \quad y_1 - a x_2 = 0, \quad y_2 - a x_1 = 0, \quad y_3 + a = 0.$$

But it is allowed to manipulate the equations and one has to keep in mind that the Study quadric equation has to be fulfilled! Substituting the first equation into the second simplifies the second equation to $y_0 - a x_3 = 0$. Therefore we have four linear equations and one quadratic equation. But the quadratic equation is essentially the Study quadric, because substituting the four linear equations into the Study quadric equation yields exactly this first equation. The 2-R chain is therefore represented by the four linear equations

$$y_0 - a x_3 = 0, \quad y_1 - a x_2 = 0, \quad y_2 - a x_1 = 0, \quad y_3 + a = 0.$$

In the kinematic image space these four equations determine a linear 3-space and all poses the 2-R chain can attain are the points of the intersection of this 3-space with the Study quadric (see also Selig (2005) and Pfurner (2006) where this method was used for deriving a fast algorithm for the inverse kinematics of general 6-R chains).

3.3 Linear Implicitization

To derive the constraint equations of a kinematic chain using geometric properties or elimination has several disadvantages. The first method needs geometric

insight into the motion of the chain and the second method introduces "spurious" solutions that have to be identified and removed from the equations. The question arises if there exists an algorithm that derives "automatically" from a parametric representation of the (allowed) kinematic chain a minimal set of implicit equations that completely describes this kinematic chain. Such an algorithm was developed in Walter and Husty (2010) and called "Linear Implicitization Algorithm" (LIA). It was shown that the parametric representation of any kinematic chain can be transformed into a set of polynomials. The basic idea of this algorithm is that first of all a kinematic chain built from only revolute and prismatic joints can be represented by a set of polynomials. Secondly one can observe that the parametric expressions have to fulfill the polynomial equations. But unfortunately the degree of the polynomials is unknown from beginning. The algorithm will be explained in detail but at first a simple example is shown to demonstrate the basic idea.

Example 9. *In this example the implicit equation of a sphere is derived. The sphere is given by its parametric expressions*

$$\begin{aligned} X &= \cos(\phi)\cos(\psi) - 3 \\ Y &= \cos(\phi)\sin(\psi) - 2 \\ Z &= \sin(\phi) - 1. \end{aligned} \tag{32}$$

Half tangent substitution yields

$$\begin{aligned} X &= \frac{(1-u^2)(1-t^2)}{(u^2+1)(t^2+1)} - 3 \\ Y &= \frac{2(1-u^2)t}{(u^2+1)(t^2+1)} - 2 \\ Z &= \frac{2u}{(u^2+1)} - 1. \end{aligned} \tag{33}$$

Actually one would have to check if a linear polynomial fulfills the parametric equation. We skip this step because we know that the implicit equation needs a quadratic polynomial. In a next step a quadratic ansatz polynomial is defined

$$Ap: \quad ax^2 + by^2 + cz^2 + dxy + exz + fyz + gx + hy + iz + j = 0. \tag{34}$$

The ten unknown coefficients of Ap have to be determined such that a substitution of the parametric expressions Eq.(33) *fulfill the equation Eq.*(34)*. Substitution of the parametric expressions Eq.*(33) *into Eq.*(34) *yields, after computing a common denominator and taking the numerator, a polynomial of degree 8 in u and t which has to be identical zero for every u and t. This can be only the case when the coefficients of every power-product in this equation vanish. Although this polynomial*

is quite large it is displayed

$$
\begin{aligned}
&4d+2f+c+2e+4a-2h-2g-i+4b+j+\\
&(4a+4b+c+4d+2e+2f-2g-2h-i+j)t^4u^4+\\
&(-4c-4e-4f+2i)t^4u^3+(16a+8b+6c+12d+6e+4f-6g-4h-2i+2j)t^4u^2+\\
&(-4c-8e-4f+2i)t^4u+(8b+4d+2f-2h)t^3u^4+\\
&(16a+12b+2c+12d+6e+4f-6g-4h-2i+2j)t^2u^4+(-8c-12e-8f+4i)t^2u^3+\\
&(40a+8b+12c+24d+12e+8f-12g-8h-4i+4j)t^2u^2+(4i-8c-12e-8f)t^2u+\\
&(8b+8d+2f-2h)tu^4+(-8b-4d-2f+2h)t+\\
&(16a+4b+c+8d+4e+2f-4g-2h-i+j)u^4+(-4c-8e-4f+2i)u^3+\\
&(16a+8b+6c+12d+6e+4f-6g-4h-2i+2j)u^2+(-4c-4e-4f+2i)u+\\
&(16a+4b+c+8d+4e+2f-4g-2h-i+j)t^4+(-8b-8d-2f+2h)t^3+\\
&(16a+12b+2c+12d+6e+4f-6g-4h-2i+2j)t^2-4ft^3u^3+4dt^3u^2+\\
&4ft^3u-4ftu^3-4dtu^2+4ftu=0. \qquad (35)
\end{aligned}
$$

One can observe that each coefficient of the power products yields linear equations in the ten unknowns a,b,c,d,e,f,g,h,i,j. As there are 25 equations we have a highly overconstrained system of linear equations. Nevertheless this system has to have a unique solution which reads

$$a=1,b=1,c=1,d=0,e=0,f=0,g=6,h=4,i=2,j=13, \qquad (36)$$

and from this solution we obtain the implicit equation of the sphere

$$x^2+y^2+z^2+6x+4y+2z+13=0. \qquad (37)$$

An easy computation shows that the parametric expressions Eq.(32) *and Eq.*(33) *fulfill the implicit equation Eq.*(37)*. This simple example already shows the disadvantage of this algorithm. The linear system will always be overconstrained and can become very large even for relatively simple chains.*

Remark 2. *In case the linear system has no solution or only the trivial solution one has to repeat the algorithm with an ansatz polynomial of higher degree.*

We proceed now to explain the LIA for a general kinematic chain. In the first step of the algorithm one has to compute the forward transformation of the kinematic chain to obtain a parametric expression.

If the relative position of two rotation axes is described by the usual Denavit-Hartenberg parameters (α_i, a_i, d_i) the coordinate transformation between the coor-

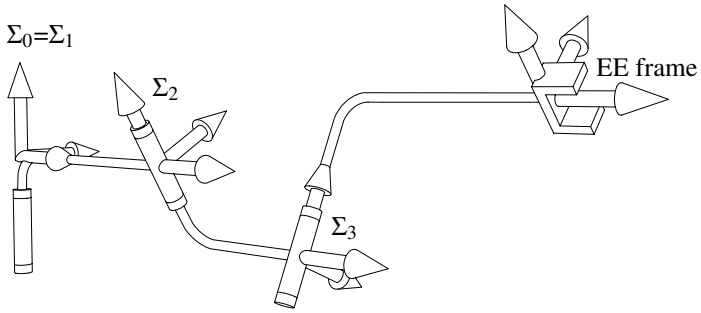


Figure 13. Canonical 3R-chain

dinate systems attached to the rotation axes is given by

$$\mathbf{G}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i & 1 & 0 & 0 \\ 0 & 0 & \cos(\alpha_i) & -\sin(\alpha_i) \\ d_i & 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix}. \tag{38}$$

Using this transformation we assume the axes of an *nR*-chain to be in a canonical start position, where all the axes are parallel to a plane, the first rotation axis is the z-axis of the base coordinate system and the x-axis is the common normal of first and second revolute joint. A simple consideration shows that this is always possible and no restriction of generality (Pfurner (2006), Husty et al. (2007b)). As shown in Fig.13 the motion axes are always the z-axes of the coordinate systems, they are either rotations or translations and can be written

$$\mathbf{M}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(u_i) & -\sin(u_i) & 0 \\ 0 & \sin(u_i) & \cos(u_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{M}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_i & 0 & 0 & 1 \end{bmatrix}. \tag{39}$$

Following this sequence of transformations the end-effector pose will have the following description:

$$\mathbf{D} = \mathbf{M}_1 \cdot \mathbf{G}_1 \cdot \mathbf{M}_2 \cdot \mathbf{G}_2 \cdot \cdots \cdot \mathbf{M}_n, \tag{40}$$

where n is the number of motion axes in the chain. From this parametric representation of the chain the parametric representation in the kinematic image space has to be computed using Eqs.(7) and (8). Half tangent substitution transforms the rotation angles u_i into algebraic parameters t_i and one ends up with eight parametric

equations of the form:

$$\begin{aligned} x_0 &= f_0(t_1,\dots t_n),\\ x_1 &= f_1(t_1,\dots t_n),\\ &\vdots\\ y_3 &= f_8(t_1,\dots t_n). \end{aligned} \tag{41}$$

These equations will be rational having a denominator of the form $(1+t_1^2)\cdot\ldots\cdot(1+t_n^2)$ which can be canceled because the Study parameters x_i, y_i are homogeneous. The same can be done with a possibly appearing common factor of all parametric expressions. After these allowed manipulations the simplest possible parametric representation of the kinematic chain in the kinematic image space has been found.

It is well known that there exists a one-to-one correspondence from all spatial transformations to the Study quadric which lives in $\mathbb{P}^7$. Particularly this means that a tuple of Study parameters describing a transformation is a projective point and consequently always only unique up to scalar multiples. If we have a transformation parameterized by n parameters $t_1,\dots,t_n$ we obtain by kinematic mapping a set of corresponding points in $\mathbb{P}^7$ and we ask now for the smallest variety $\mathscr{V}\in\mathbb{P}^7$ (with respect to inclusion) which contains all these points. What do we know about this variety? What can be said definitely is that its describing set of polynomials consists of homogeneous polynomials and contains $x_0y_0+x_1y_1+x_2y_2+x_3y_3$, i.e. the equation for the Study quadric S_6^2. In the following it is shown how additional equations can be computed which are necessary to describe $\mathscr{V}$. It should be noted that the minimum number of polynomials necessary to describe $\mathscr{V}$ corresponds to the number of parameters, which in turn correspond to the degrees of freedom (dof) of the kinematic chain. If the number of generic parameters is n then $m=6-n$ polynomials are necessary to describe $\mathscr{V}$. This is of course a rough statement, because different numbers can appear when special situations (e.g. redundant dofs, special design parameters) are in place.

Now we are searching for homogeneous polynomials which vanish on all points that can be obtained from the parameterization of the kinematic chain, i.e. polynomials in $x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3$ which vanish when the expressions of the parameterization are substituted. One possibility to find such polynomials is the following: A general ansatz of a homogeneous polynomial in $x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3$ with given degree n is made and then the Study parameters of the parametric representation are substituted. The resulting expression f is treated as a polynomial $f(t_1,\dots t_n)$. Due to the fact that f has to vanish for all values of the t_i, it has to be the zero polynomial. As it already was demonstrated in Example 9 all coefficients of f have to vanish. This means that, after extraction of these coefficients, one obtains a system of linear equations where the unknowns are the $\binom{n+7}{n}$ coefficients from the general ansatz. This system can be solved (assuming that the

design parameters a_i, d_i and α_i are generic) and the solution can be substituted into the ansatz. The result is an expression r describing all homogeneous polynomials of degree n which vanish on the points of $\mathscr{V}$. An important point is that if the solution of the linear system is positive dimensional, the corresponding parameters also appear in the final expression, i.e. the expression r itself is parameterized.

In the following this important part of the algorithm is explained in more detail: The simplest homogeneous ansatz polynomial would be a linear one. Therefore at first a general linear polynomial is generated:

$$ansatz: \quad C_1y_3 + C_2x_3 + C_3y_2 + C_4x_2 + C_5y_1 + C_6x_1 + C_7y_0 + C_8x_0, \qquad (42)$$

with unknown coefficients C_i. The question is now if there exist values for these unknowns such that Eq.(42) vanishes identically for points fulfilling the parametric representation of the given kinematic chain. To test this the parametric expressions Eq.(41) are substituted into Eq.(42) which yields a polynomial in t_i with coefficients in C_i and the Denavit-Hartenberg parameters a_i, d_i and α_i. We collect with respect to the powerproducts of the t_i and extract their coefficients. This yields a set of linear equations in C_i, a_i, d_i and α_i. The number of equations depends on the particular design of the chain. In general the system will consist of more equations than unknowns because in general there are more powerproducts than unknowns C_i. This does not mean that there is no solution, because the equations can be dependent. Already in the Example 9 we have seen that they have to be dependent, at least if the degree of the ansatz polynomial is high enough, because the constraint variety will have some algebraic degree.

In the next step the system of linear equations is solved. If this system has a solution, then there exists a linear constraint polynomial which is an element of the set of polynomials describing $\mathscr{V}$. If this system has no solution, which means only the null vector solves it, one has to proceed to degree two. A general ansatz polynomial of degree two is created and then one has to follow the same steps as above. Note that a general quadratic polynomial in Study coordinates has 36 coefficients. Depending on the design of the chain one obtains a system of linear equations in 36 unknowns. For many kinematic chains the second step already yields solutions. For example a chain consisting of a universal joint and a spherical joint (as it is the case for one leg of a Stewart-Gough platform, when the actuated P-joint is locked) the second step comes immediately up with the quadratic constraint equation for this leg. This case will be shown in the next example explicitly.

Note that as many polynomials have to be obtained as the chain has constraints c. From this follows that the algorithm has to be continued increasing the degree of the ansatz polynomial until the number of necessary equations is obtained.

In the following several examples will be presented. We follow the general outline of the algorithm and we will only show how the constraint equations of canonical chains are found.

Example 10 (Canonical leg of a Stewart-Gough platform (UPS-chain))**.** *It is well known that the Stewart-Gough parallel manipulator (SGP) consists of a base and a platform connected by six identical legs (Husty (1996)). Each of these legs is a serial chain that can be modeled as a UPS serial linkage. To compute the direct kinematics the actuated prismatic joint is considered to be locked. The resulting canonical linkage therefore consists of five revolute joints and the forward kinematics of this chain is described by the coordinate transformations:*

$$\mathbf{D} = \mathbf{M}_1 \cdot \mathbf{G}_1 \cdot \mathbf{M}_2 \cdot \mathbf{G}_2 \cdot \mathbf{M}_3 \cdot \mathbf{G}_3 \cdot \mathbf{M}_4 \cdot \mathbf{G}_4 \cdot \mathbf{M}_5. \tag{43}$$

The necessary entries of the coordinate transformation matrices are displayed in Table 10. By computing the forward kinematics Eq.(43) *and then transforming to*

	α_i	a_i	d_i
$\mathbf{G}_1$	$\frac{\pi}{2}$	0	0
$\mathbf{G}_2$	0	L	0
$\mathbf{G}_3$	$\frac{\pi}{2}$	0	0
$\mathbf{G}_4$	$\frac{\pi}{2}$	0	0

Table 1. Denavit-Hartenberg parameters of the UPS-chain

Study parameters using Eqs.(7) *and* (8) *and performing half tangent substitution one will arrive at the parametric representation of the variety representing this linkage in* $\mathbb{P}^7$*:*

$$\begin{aligned}
x_0 = {} & -1+t_5t_1-t_5t_2-t_5t_1t_2t_3+t_2t_5t_1t_4-t_1t_4-t_5t_4-t_5t_3+t_4t_1t_2t_3+t_4t_2t_3t_5-t_1t_2- \\
& t_4t_3-t_4t_2+t_4t_1t_3t_5-t_1t_3+t_2t_3 \\
x_1 = {} & -t_4t_1t_2t_3-t_5t_1t_2t_3-t_2t_5t_1t_4-t_1t_2-t_4t_1t_3t_5-t_1t_3+t_1t_4+t_5t_1+t_4t_2t_3t_5-t_2t_3 \\
& -t_4t_2+t_5t_2-t_4t_3+1+t_5t_3-t_5t_4 \\
x_2 = {} & t_1+t_2-t_1t_2t_3-t_4t_1t_2+t_1t_4t_2t_3t_5-t_4+t_5t_1t_2+t_3+t_2t_5t_4+t_4t_2t_3+t_5t_2t_3-t_4t_1t_3 \\
& -t_5+t_5t_1t_3-t_5t_1t_4+t_4t_3t_5 \\
x_3 = {} & -t_1+t_2+t_1t_2t_3-t_5t_1t_3-t_4t_1t_2+t_1t_4t_2t_3t_5+t_4-t_5t_1t_2+t_3-t_5t_1t_4-t_4t_2t_3- \\
& t_4t_1t_3-t_5+t_5t_2t_3-t_2t_5t_4-t_4t_3t_5
\end{aligned}$$

$$
\begin{aligned}
y_0 = & -\frac{1}{2}L(-1+t_1t_4-t_4t_1t_3t_5+t_4t_1t_2t_3+t_1t_2+t_5t_1+t_4t_3+t_5t_1t_2t_3-t_4t_2+t_2t_5t_1t_4- \\
& t_1t_3-t_2t_3+t_5t_2+t_4t_2t_3t_5-t_5t_3+t_5t_4) \\
y_1 = & -\frac{1}{2}L(-1+t_1t_4-t_4t_1t_3t_5-t_5t_3+t_4t_1t_2t_3-t_1t_2-t_4t_3-t_5t_1t_2t_3+t_4t_2+t_1t_3- \\
& t_2t_3-t_4t_2t_3t_5-t_5t_1+t_5t_2-t_5t_4+t_2t_5t_1t_4) \\
y_2 = & \frac{1}{2}L(t_1-t_2+t_1t_2t_3+t_5t_1t_3-t_4t_1t_2+t_1t_4t_2t_3t_5+t_4-t_5t_1t_2+t_3+t_5t_1t_4+ \\
& t_4t_2t_3+t_4t_1t_3-t_5-t_5t_2t_3+t_2t_5t_4-t_4t_3t_5) \\
y_3 = & -\frac{1}{2}L(-t_1-t_2-t_1t_2t_3-t_5t_1t_3-t_4t_1t_2+t_1t_4t_2t_3t_5-t_4+t_5t_1t_2+t_3+t_5t_1t_4- \\
& t_4t_2t_3+t_4t_1t_3-t_5-t_5t_2t_3-t_2t_5t_4+t_4t_3t_5). \qquad (44)
\end{aligned}
$$

As this chain has five parameters we can expect one equation that will describe the constraint variety $\mathscr{V}$ together with the equation for S_6^2. One may see immediately that an elimination process will be very tedious, maybe hopeless. Definitely the elimination will blow up the degree of the resulting equations. We will show that the proposed algorithm yields a result after two steps: In the first step the parametric equations Eq.(44) *are substituted into the linear ansatz equation Eq.*(42)*. The resulting equation is collected with respect to the monomials in t_i. We obtain an expression with 32 terms. Only the beginning and the end of this expression is displayed:*

$$
\begin{aligned}
& (C_3L+C_1L+2C_4-2C_2)t_1+(-C_7L+2C_6+C_5L+2C_8)t_4t_2t_3t_5 \qquad (45) \\
& +(C_7L+C_5L+2C_8-2C_6)t_4t_1t_3t_5+\ldots+(C_3L+C_1L+2C_4-2C_2)t_4t_2t_3=0.
\end{aligned}
$$

It turns out that this system of 32 linear equations in 8 unknowns has no solution. Therefore no linear equation describing the constraint variety exists. In the next step we build a general second order equation and proceed as before. Now we obtain a linear system with 234 equations in 32 unknowns. This system is solved and it yields a two dimensional solution vector, which is back substituted into the quadratic ansatz equation:

$$
(y_0^2+y_1^2+y_2^2+y_3^2-\frac{1}{4}L^2(x_0^2+x_1^2+x_2^2+x_3^2))\lambda+(x_0y_0+x_1y_1+x_2y_2+x_3y_3)\mu=0. \qquad (46)
$$

This surprisingly simple result shows that the constraint variety of the canonical leg of the Stewart-Gough platform is described by the Study quadric equation and the second quadratic equation in Eq.(46) *and linear combinations of these two equations.*

One question remains to be solved: how do we obtain the constraint equations of a chain in general position from the constraint equations of a kinematic chain in

canonical position? The answer to this question was given in Pfurner (2006) and will be recalled briefly here. Suppose that $\alpha\colon \mathbf{x} \mapsto \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{a}$ is a Euclidean displacement. The vectors $\mathbf{x}$ and $\mathbf{y}$ are elements of $\mathbb{R}^3$ but in kinematics it is advantageous to consider them as elements of two distinct copies of $\mathbb{R}^3$, called the *moving space* and the *fixed space*. The description of α in Study parameters depends on the choice of coordinate frames – *moving frame* and *fixed* or *base frame* – in both spaces. In kinematics, the moving frame is the space attached to a mechanism's output link, and the fixed space is the space where the mechanism itself is positioned. Both types of transformations induce transformations of the Study quadric and thus impose a geometric structure on $\mathbb{P}^7$. Kinematic mapping is constructed such that these transformations act linearly on the Study parameters (that is, they are projective transformations in $\mathbb{P}^7$). We are going to compute their coordinate representations.

Consider a Euclidean displacement described by a four by four transformation matrix $\mathbf{X}$, as in Eq.(3). It maps a point $(1,\mathbf{a})^T$ to $(1,\mathbf{a}')^T = \mathbf{X}\cdot(1,\mathbf{a})^T$. Now we change coordinate frames in fixed and moving space and compute the matrix $\mathbf{Y}$ such that $(1,\mathbf{b}')^T = \mathbf{Y}\cdot(1,\mathbf{b})^T$ is the representation of the displacement in the new fixed coordinate frame and the *old* moving coordinate frame. This is slightly different from the typical change of coordinates known from linear algebra where one describes the new transformation in terms of new coordinates in *both* spaces but more suitable for application in kinematics, in particular for describing the position of the end effector tool or for concatenation of different joint displacements in kinematic chains. When the changes of coordinates in fixed and moving frame are described by

$$(1,\mathbf{a})^T = \mathbf{M}\cdot(1,\mathbf{b})^T, \quad (1,\mathbf{b}')^T = \mathbf{F}\cdot(1,\mathbf{a}')^T, \tag{47}$$

we have $\mathbf{Y} = \mathbf{F}\cdot\mathbf{X}\cdot\mathbf{M}$. Denote now by $\mathbf{y}$, $\mathbf{x}$, $\mathbf{f} = [f_0,\ldots,f_7]^T$ and $\mathbf{m} = [m_0,\ldots,m_7]^T$ the corresponding Study vectors. Straightforward computation (see Pfurner (2006)) yields

$$\mathbf{y} = \mathbf{T}_f\mathbf{T}_m\mathbf{x}, \quad \mathbf{T}_m = \begin{bmatrix}\mathbf{A} & \mathbf{O}\\ \mathbf{B} & \mathbf{A}\end{bmatrix}, \quad \mathbf{T}_f = \begin{bmatrix}\mathbf{C} & \mathbf{O}\\ \mathbf{D} & \mathbf{C}\end{bmatrix}, \tag{48}$$

where

$$\mathbf{A} = \begin{bmatrix} m_0 & -m_1 & -m_2 & -m_3\\ m_1 & m_0 & m_3 & -m_2\\ m_2 & -m_3 & m_0 & m_1\\ m_3 & m_2 & -m_1 & m_0\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} m_4 & -m_5 & -m_6 & -m_7\\ m_5 & m_4 & m_7 & -m_6\\ m_6 & -m_7 & m_4 & m_5\\ m_7 & m_6 & -m_5 & m_4\end{bmatrix}, \tag{49}$$

$$\mathbf{C} = \begin{bmatrix} f_0 & -f_1 & -f_2 & -f_3\\ f_1 & f_0 & -f_3 & f_2\\ f_2 & f_3 & f_0 & -f_1\\ f_3 & -f_2 & f_1 & f_0\end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} f_4 & -f_5 & -f_6 & -f_7\\ f_5 & f_4 & -f_7 & f_6\\ f_6 & f_7 & f_4 & -f_5\\ f_7 & -f_6 & f_5 & f_4\end{bmatrix}, \tag{50}$$

and **O** is the four by four zero matrix.

Example 11 (Continuation of Example 10)**.** *The computation of the constraint equations of a kinematic chain in general position from the canonical equations will be demonstrated for the SGP manipulator. To move the canonical leg from the best adapted position, which has the origin of the fixed frame in the intersection point of axes of the U-joint and the origin of the moving frame in the center of the spherical joint, into a general position, two translations, one in the fixed and one in the moving frame, are needed. Both transformation matrices of Eq.*(47) *become*

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A_1 & 1 & 0 & 0 \\ B_1 & 0 & 1 & 0 \\ C_1 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -a_1 & 1 & 0 & 0 \\ -b_1 & 0 & 1 & 0 \\ -c_1 & 0 & 0 & 1 \end{bmatrix}. \tag{51}$$

Transforming into Study coordinates and constructing the Matrices $\mathbf{T}_f$ *and* $\mathbf{T}_m$ *yields*

$$\mathbf{T}_m\mathbf{T}_f = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & -2a1+2A_1 & -2b_1+2B_1 & -2c_1+2C_1 & 4 & 0 & 0 & 0 \\ 2a_1-2A_1 & 0 & 2c_1+2C_1 & -2b_1-2B_1 & 0 & 4 & 0 & 0 \\ 2b_1-2B_1 & -2c_1-2C_1 & 0 & 2a_1+2A_1 & 0 & 0 & 4 & 0 \\ 2c_1-2C_1 & 2b_1+2B_1 & -2a_1-2A_1 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}. \tag{52}$$

We denote both Study vectors in Eq.(48) *by* $\mathbf{x} = (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3)^T$ *and* $\mathbf{y} = (x'_0, x'_1, x'_2, x'_3, y'_0, y'_1, y'_2, y'_3)^T$ *and realize that we have to use the inverse of the transformation Eq.*(52) *to obtain expressions in the coordinates of* $\mathbf{x}$

$$\mathbf{x} = (\mathbf{T}_m\mathbf{T}_f)^{-1}\mathbf{y}$$

so that these expressions can be substituted into the canonical constraint equation of the leg in optimal position with respect to both coordinate systems. One arrives

at the equation

$$\begin{aligned}
&\left((a_1-A_1)x'_1+(b_1-B_1)x'_2+(c_1-C_1)x'_3+2y'_0\right)^2+\\
&\left((-a_1+A_1)x'_0+(-c_1-C_1)x'_2+(b_1+B_1)x'_3+2y'_1\right)^2+\\
&\left((-b_1+B_1)x'_0+(c_1+C_1)x'_1+(-a_1-A_1)x'_3+2y'_2\right)^2+\\
&\left((-c_1+C_1)x'_0+(-b_1-B_1)x'_1+(a_1+A_1)x'_2+2y'_3\right)^2-\\
&\frac{1}{4}L^2\left(4{x'_0}^2+4{x'_1}^2+4{x'_2}^2+4{x'_3}^2\right)=0. \qquad (53)
\end{aligned}$$

Eq.(53) *is the constraint equation of kinematic chain corresponding to leg of the SGP in general position. It is obvious that the equation is still quadratic and additionally it can be seen that all* y'^2_i *are free of design parameters. An analogous fact to the planar case in Example 3.*

Example 12 (3-R chain)**.** *In this example the constraint variety of a 3-R chain is discussed. This variety is interesting because it has been used in a semi parametric form in Husty et al. (2007a), Husty et al. (2007b), Pfurner (2006) to efficiently compute the inverse kinematics of general 6-R serial manipulators. In this case we have to expect three constraint equations because the dimension of the constraint variety must be three.*

As in the two previous cases the linear ansatz does not give a constraint equation. On the other hand the quadratic ansatz yields nine quadratic equations. All nine equations fulfill the parametric equations of this chain. Any three of these nine equations plus the equation for S_6^2 *(which is contained in the set) can be taken for further computations. It is guaranteed that the variety generated by them will contain all points that correspond to poses of the endeffector of the 3-R chain. But it might be that the variety is bigger than necessary. This means that the other six quadratic equations can (or better should) be taken for solution verification.*

More details on LIA can be found in Walter and Husty (2010), where especially some remarks are made to special situations that can occur during the algorithm.

3.4 Singularities in serial chains

Algebraic methods are of course not restricted to the analysis of parallel manipulators. As the basic element of this methods are kinematic chains, serial manipulators can be analyzed with the same tool. But one has to be aware that serial and parallel manipulators in some sense behave dual to the basic kinematic tasks. As we have seen in the last subsection the forward kinematics of a kinematic chain is straight forward (Eq.(40)). The inverse kinematics of serial manipulators is more complicated, but using the algebraic methods this task even for the most general

6-R robot can be solved very efficiently (Pfurner (2006)). The algorithm is so efficient, that a C++ implementation solves almost all cases in real time (Angerer (2016)). The application of the algebraic methods for serial manipulators is show in the computation of all singularities of an industrial robot.

Example 13 (Singularities of a 6-R robot)**.** *The geometry of the robot under investigation is given by the following list of DH-parameters.*

$$a_1 = 0, a_2 = 85/2, a_3 = 39, a_4 = 0, a_5 = 0, a_6 = 0, \alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0,$$
$$\alpha_4 = -1, \alpha_5 = 1, \alpha_6 = 0, d_1 = 9, d_2 = 0, d_3 = 0, d_4 = 10, d_5 = 9, d_6 = 8. \quad (54)$$

Note that the angles are given by their algebraic values.The most efficient way to compute the Jacobian of a serial manipulator is given by the fact that its columns contain the Plücker coordinates of the instantaneous position of its axes. To perform this operations one needs the transformation matrix that transforms lines in a Euclidean transformation. Let $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$ *and* $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)^T$ *be the Plücker coordinates of two lines, then the transformation of* $\mathbf{p}$ *with a transformation as in Eq.*(2) *is given by*

$$\mathbf{q} = \begin{bmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{a}^{\times}\mathbf{M}_R & \mathbf{M}_R \end{bmatrix} \mathbf{p}, \quad (55)$$

where $\mathbf{a}^{\times}$ *is the skew symmetric operator belonging to the vector* $\mathbf{a} = (a_1, a_2, a_3)^T$*:*

$$\mathbf{a}^{\times} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}.$$

Performing this operation for every axis of the manipulator yields the columns Y_i *of the Jacobian matrix* $\mathbf{J}$*. The singularities of the manipulator are the given by* $\det\mathbf{J} = 0$*. Computing this determinant for the manipulator given by the DH-parameters Eq.*(54) *yields*

$$(t_1^2+1)^5(t_2^2+1)^3(t_4^2+1)(t_3^2+1)t_3t_5(7t_2^2t_3^2t_4^2+36t_2^2t_3^2t_4+36t_2^2t_3t_4^2+$$
$$36t_2t_3^2t_4^2+7t_2^2t_3^2+163t_2^2t_4^2+312t_2t_3t_4^2-7t_3^2t_4^2-36t_2^2t_3-36t_2^2t_4-$$
$$36t_2t_3^2-144t_2t_3t_4-36t_2t_4^2-36t_3^2t_4-36t_3t_4^2+163t_2^2+312t_2t_3-7t_3^2-$$
$$163t_4^2+36t_2+36t_3+36t_4-163)=0. \quad (56)$$

The t_i *in this equation are the algebraic values of the joint parameters in the six axes. The first four factors in the equation can be canceled because they can never become real. Then one can observe that first and sixth rotation angle do not contribute to the singularities. This well know fact applies to all 6-R robots.* $t_3 = 0$

means that the second, third and fourth axes are parallel, $t_5 = 0$ means that the fourth and fifth axes are parallel. In both cases the manipulator is in a singular pose. The remaining equation is of degree 6 in t_2, t_3 and t_4 describes all remaining singular poses of the manipulator. As it contains only three input variables it even can be visualized (Fig.14).

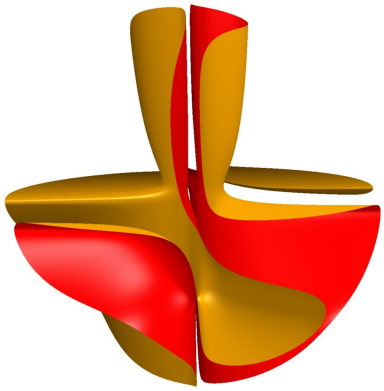


Figure 14. Singularity surface of a 6-R robot with DH parameters Eq.(54)

4 Algebraic Basics

In this introduction only the most important definitions and theorems from algebraic geometry are recalled. It should serve the interested reader as a first introduction to this topic. The main emphasis is put on those issues of algebraic geometry that have turned out to be of a certain relevance to the kinematic analysis of mechanisms and robots. This section is based on the book *"Ideals, Varieties, and Algorithms"* by Cox et al. (2007), which is an excellent introduction to algebraic geometry. More detailed descriptions and also the proofs for the theorems can be found there.

Examples were computed in *Maple 16* mostly using the packages `Groebner` and `PolynomialIdeals`. The first package contains the low-level commands, the second package is newer and contains the more sophisticated ones. There are of course also other software packages like e.g. *Mathematica*, *Singular* or *Macaulay 2* that could be used for such computations. In most of the examples the Maple output is not shown, the reader is advised to test the Maple commands by typing the displayed commands into the system.

In the following all algebraic equations are polynomials in the ring $\mathbf{K}[\underline{x}] = \mathbf{K}[x_1,\ldots,x_n]$ where $\mathbf{K}$ is a field like $\mathbb{Q}$ (the rational numbers) or $\mathbb{C}$ (the complex numbers).

4.1 Ideals and Affine Varieties

At first polynomial ideals are defined which are the basic objects for everything else.

Definition 1. *A set $I \subseteq \mathbf{K}[\underline{x}]$ is called an **ideal** if the following conditions are fulfilled:*

- $\forall f,g \in I : f+g \in I$,
- $\forall f \in I$ *and* $\forall h \in \mathbf{K}[\underline{x}] : hf \in I$.

It follows that almost all ideals are infinite sets of polynomials and cannot be written down as a whole. The sole exception is the ideal $\{0\}$ which is also a proper subset resp. subideal of all other ideals because 0 is contained in every ideal. There is also an ideal which is a proper superset resp. superideal of them, namely the ideal which contains the constant polynomial 1 and with it all polynomials of $\mathbf{K}[\underline{x}]$.
Using Definition 1 it is possible to define the ideal generated by a set of given polynomials $f_1,\ldots,f_s$.

Definition 2. *Let $f_1,\ldots,f_s$ be polynomials in $\mathbf{K}[\underline{x}]$. Then the set*

$$\langle f_1,\ldots,f_s\rangle = \{g \in \mathbf{K}[\underline{x}] : g = \sum_{i=1}^{s} h_i f_i \text{ and } h_1,\ldots,h_s \in \mathbf{K}[\underline{x}]\}$$

*is the **ideal generated by** $f_1,\ldots,f_s$.*

The ideal generated be the given polynomials is the set of all combinations of these polynomials using coefficients from $\mathbf{K}[\underline{x}]$. The polynomials $f_1,\ldots,f_s$ form a so called *basis* of the ideal. Due to the fact that the same ideal can be generated by another set of polynomials, such a basis is not unique. Furthermore the ideal is obviously *finitely generated* and it can be shown that every ideal of $\mathbf{K}[\underline{x}]$ can be generated by a finite set of polynomials (Hilbert Basis Theorem). So the two special ideals mentioned above can be written as $\langle 0\rangle$ and $\langle 1\rangle$.

Example 14. *A circle shall be intersected with an ellipse where the corresponding algebraic equations are $f_1 = (x_1-1)^2+(x_2-2)^2-4=0$ and $f_2 = x_1^2+3x_2^2-5=0$. To formulate this problem using Maple one has to type in the following to define the corresponding ideal:*

```
with(PolynomialIdeals);
```

```
I1:=<(x[1]-1)^2+(x[2]-2)^2-4,x[1]^2+3*x[2]^2-5>;
```

All commands in the package `Groebner` *allow to use the notation* $[\dots]$ *instead of* $<\cdots>$.

Remark 3. *In this newly introduce language the three constraint equations* h_1, h_2, h_3 *of Example 3 (Eq.*(21)*) constitute an ideal. Definition 1 guarantees that the operations that were used to derive the univariate polynomial (e.g. taking the differences of two of the given equations) result in polynomials that are contained in the ideal.*

As it was mentioned above such a basis is not unique. For example the ideals $\langle f_1, f_2\rangle$ and $\langle f_1 - f_2, f_2\rangle$ are completely the same. But how can this be found out for two given ideals, if the are equal or not?

Two ideals I and J are equal if each element of I is contained in J and vice versa. It is sufficient to test if the basis of one ideal is contained in the other, and vice versa. To find out if a given polynomial is a member of an ideal it is necessary to test if the polynomial can be written as a combination of the ideal's basis. How this can be done in a systematic way will be explained later.
Before that affine varieties are introduced.

Definition 3. *For a given ideal* $I = \langle f_1, \dots, f_s\rangle \subseteq \mathbf{K}[\underline{x}]$ *the set*

$$\mathbf{V}(I) = \{(a_1, \dots, a_n) \in \mathbf{K}^n : f_i(a_1, \dots, a_n) = 0 \ for\ all\ 1 \le i \le s\} \subseteq \mathbf{K}^n$$

is called the ***affine variety*** *of the ideal* I.

For each ideal $I = \langle f_1, \dots, f_s\rangle$ there exists a unique variety $\mathbf{V}(I)$ which is the set of all solutions of the polynomials equations $f_1 = 0, \dots, f_s = 0$, the so called *vanishing set*. It follows immediately that all bases of the ideal describe the same variety. In general the variety of an ideal is the more interesting thing, not the ideal itself, because the variety is exactly the set of solutions of the input equations $f_1, \dots, f_s$. It has to be mentioned explicitly that the variety does not contain information about the multiplicity of solutions. It is just a set of points in $\mathbf{K}[\underline{x}]$, nothing more.

Two special varieties are $\emptyset$ and $\mathbf{K}^n$ which are the vanishing sets of the ideals $\langle 1\rangle$ and $\langle 0\rangle$ which appeared earlier.

Example 15. *A circle with center* $(0,0)$ *and a line are given by* $x_1^2 + x_2^2 - 1 = 0$ *and* $x_1 + x_2 - 1 = 0$. *Then the ideal generated by these two equations is given by* $I = \langle x_1^2 + x_2^2 - 1, x_1 + x_2 - 1\rangle = \langle f_1, f_2\rangle$ *and the corresponding variety is* $\{(1,0),(0,1)\}$.

The affine variety belonging to the first equation is the set of points that are on the circle and those of the second equation is the set of points that are on the

line. The affine variety belonging to the generated ideal I is the set of points that belong to both varieties, i.e. the set of intersection points (Fig.15). A third polynomial of the ideal is constructed $f_3 = 3f_1 - 10f_2^3 = -10x_1^3 - 30x_1^2x_2 - 30x_1x_2^2 - 10x_2^3 + 33x_1^2 + 60x_1x_2 + 33x_2^2 - 30x_1 - 30x_2 + 7$. *In Fig.17 one can see that all three polynomials have the same solution set, i.e. they constitute the same affine variety.*

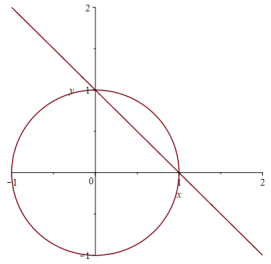


Figure 15. Affine varieties belonging to f_1, f_2

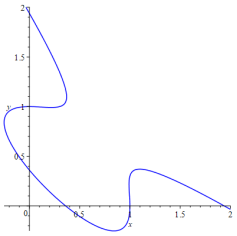


Figure 16. Affine variety belonging to f_3

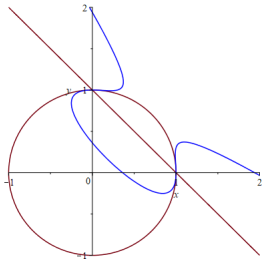


Figure 17. Affine varieties belonging to f_1, f_2, f_3

It is also possible that different ideals describe the same variety. This is related to the fact that solutions can appear with higher multiplicities.

Example 16. *The following polynomial ideals I and I' are given, each by two possible bases.*

$$\begin{aligned} I &= \langle x_1^2 + x_2^2 - 1, x_1 + x_2 - 1\rangle = \langle x_2^2 - x_2, x_1 + x_2 - 1\rangle, \\ I' &= \langle (x_1^2 + x_2^2 - 1)^2, x_1 + x_2 - 1\rangle = \langle x_2^4 - 2x_2^3 + x_2^2, x_1 + x_2 - 1\rangle. \end{aligned}$$

It can easily be seen that $\mathbf{V}(I) = \mathbf{V}(I') = \{(1,0),(0,1)\}$ *but the ideals are not the same because* $x_2^2 - x_2$ *cannot be written as a combination of* $x_2^4 - 2x_2^3 + x_2^2$ *and* $x_1 + x_2 - 1$.

To test if two ideals describe the same variety *radicals* are introduced.

Definition 4. *Let* $I \subseteq \mathbf{K}[\underline{x}]$ *be an ideal. The set*

$$\sqrt{I} := \{f \in \mathbf{K}[\underline{x}] : \exists\, m \in \mathbb{N}, m \geq 1 \text{ with } f^m \in I\}$$

is called the ***radical*** *of I.*

The computation of the radical of an ideal I can be seen as reducing I down to the most important things, relevant for its vanishing set by taking out multiplicities. In the example above the radical is the same: $\sqrt{I} = \sqrt{I'} = I$.

Example 17. *An ideal* $I = \langle (2x_1 - x_2 - 2)x_1, (2x_1 - x_2 - 2)x_2^2 \rangle$ *is defined. Its vanishing set* $\mathbf{V}(I)$ *is the line described by* $2x_1 - x_2 - 2$ *and the isolated point* $(0,0)$*. To be exact, there are two copies of the point* $(0,0)$ *when multiplicities are taken into account.*
Computation of the radical using Maple leads to the slightly simpler ideal $\sqrt{I} = \langle -(2x_1 + x_2)(2x_1 - x_2 - 2), -(2x_1 - x_2 - 2)x_1 \rangle$ *which has the same vanishing set, but now the point* $(0,0)$ *does not appear with higher multiplicity. The code for Maple is the following:*

```
with(PolynomialIdeals);

I1:=<(2*x[1]-x[2]-2)*x[1], (2*x[1]-x[2]-2)*x[2]^2>;

rad:=Radical(I1);
```

Next some operations are given which can be applied to varieties.

Definition 5. *Let* $I = \langle f_1, \ldots, f_s \rangle$ *and* $J = \langle g_1, \ldots, g_t \rangle$ *be ideals with corresponding varieties* $V = \mathbf{V}(I)$ *and* $W = \mathbf{V}(J)$*. Then the* ***union*** *and* ***intersection*** *of V and W can be described as follows:*

$$\begin{aligned} V \cap W &= \mathbf{V}(\langle f_1, \ldots, f_s, g_1, \ldots, g_t \rangle), \\ V \cup W &= \mathbf{V}(\langle f_i g_j : 1 \leq i \leq s, 1 \leq j \leq t \rangle). \end{aligned}$$

The first equality is quite clear, if one is searching for the solutions two systems have in common, the equations are combined and the resulting system is examined. The second equality can be used to construct varieties which are a composition of simpler varieties.

Example 18. *Two very simple varieties are given to show what happens, when varieties are intersected or joined.*

$$V_1 = \mathbf{V}(\langle x_1 \rangle), \quad V_2 = \mathbf{V}(\langle x_2 \rangle),$$

$$\begin{aligned} V_1 \cap V_2 &= \mathbf{V}(\langle x_1, x_2 \rangle) = \{(0,0)\}, \\ V_1 \cup V_2 &= \mathbf{V}(\langle x_1 x_2 \rangle). \end{aligned}$$

What happens when two varieties are joined where one is a subset of the other variety?

$$V_1 \cup \mathbf{V}(\langle x_1, x_2 \rangle) = \mathbf{V}(\langle x_1^2, x_1 x_2 \rangle).$$

It can easily be seen that the vanishing set is the same, but when multiplicities are taken into account the point $(0,0)$ *appears twice.*
The code for Maple and the corresponding commands reads:

```
with(PolynomialIdeals);

J1:=<x[1]>;  J2:=<x[2]>;

J_union:=Add(J1,J2);

J_inters:=Multiply(J1,J2);

Multiply(J1,J_union);
```

Now we go back to the different ways to generate an ideal. As already mentioned there are lots of different bases which describe the same ideal, e.g. the ideals

$$\begin{aligned} I &= \langle f_1, f_2, f_3 \rangle \subseteq \mathbf{K}[x_1, x_2, x_3, x_4], \\ I' &= \langle f_1, x_2^2 f_1 - f_2, 2 f_3 + (x_3 + 5) f_2 - f_1 \rangle. \end{aligned}$$

It can easily be verified that each combination of the generators of I' can be written as a combination of generators of I and vice versa. But what to do when the ideals are more complex? It is necessary to have a systematic way for testing if a polynomial is a combination of some other polynomials. Therefore the concept of *multivariate division with remainder* is introduced, which is similar to the well known *univariate division with remainder* which will be discussed first.

Theorem 1. *Let $f, g \in \mathbf{K}[x_1]$ be univariate polynomials with $g \neq 0$. Then there exist unique polynomials q and r such that*

$$f = qg + r$$

with either $r = 0$ or $deg(r) < deg(g)$.

In the corresponding division algorithm an appropriate multiple of g is subtracted from f such that the monomial with highest degree is cancelled from f. This procedure is repeated until the remainder is either 0 or has degree less than $deg(g)$.

Example 19. *The division algorithm is demonstrated using the two polynomials $f = 2x^2 - 3x - 7$ and $g = x + 5$:*

$$\begin{aligned} f - \boldsymbol{2x}\, g &= 2x^2 - 3x - 7 - 2x(x+5) = -13x - 7, \\ (-13x - 7) - \boldsymbol{(-13)} g &= -13x - 7 + 13x + 65 = \boldsymbol{58}. \end{aligned}$$

It follows that $q = 2x - 13$ and $r = 58$. To get these results with Maple the commands would be

```
quo(f,g,x);  rem(f,g,x);
```

The process stops when the highest monomial of the remainder is not divisible by the highest monomial of g. So in the univariate case the degree of monomials can be seen as a natural *order* on the set of monomials, which guides the user trough the algorithm.

In the following different *termorders* are introduced which allow ordering of the monomials of a multivariate polynomial. With these termorders an analog algorithm to the univariate case can be defined for multivariate polynomials.

Definition 6. *Let* $\underline{x}^{\alpha} = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ *and* $\underline{x}^{\beta} = x_1^{\beta_1} \dots x_n^{\beta_n}$ *be monomials in* $\mathbf{K}[\underline{x}] = \mathbf{K}[x_1, \dots, x_n]$. *To order these monomials a* ***monomial ordering*** *or* ***termorder*** $>_{\underline{x}}$ *on the set of monomials in* $\mathbf{K}[\underline{x}]$ *is defined by an ordering* $>$ *on the n-tuples* $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$ *which has to fulfill the following conditions:*

- $>$ *is a total ordering on* $\mathbb{Z}^n_{\geq 0}$,
- *if* $\alpha > \beta$ *and* $\gamma \in \mathbb{Z}^n_{\geq 0}$, *then* $\alpha + \gamma > \beta + \gamma$,
- *every nonempty subset of* $\mathbb{Z}^n_{\geq 0}$ *has a smallest element under* $>$.

If such an ordering $>$ *on* $\mathbb{Z}_{\geq 0}$ *is given the monomials are ordered using the following equivalence:*

$$\underline{x}^{\alpha} >_{\underline{x}} \underline{x}^{\beta} \Longleftrightarrow \alpha > \beta.$$

So monomials are ordered by comparing the ordered n-tuples constructed from the powers of each variable. Next the most important termorderings are given.

Definition 7. ***(Lexicographic Order)*** *Let* $\alpha = (\alpha_1, \dots, \alpha_n)$ *and* $\beta = (\beta_1, \dots, \beta_n)$ *be elements of* $\mathbb{Z}^n_{\geq 0}$. *We define* $\alpha >_{lex} \beta$ *if the leftmost nonzero entry of the vector-difference* $\alpha - \beta \in \mathbb{Z}^n$ *is positive.*

The Maple keyword for this ordering is `plex`, e.g. a possible lexicographic termorder for polynomials containing the unknowns $\{x_1, x_2, x_3\}$ could be `plex(x[3], x[1],x[2])`.

Example 20. *How this ordering looks like for monomials in* $\mathbf{K}[x_1, x_2]$ *can be seen in the following. It is a sketch how the set of all monomials is ordered, first the 2-tuples are given, then the corresponding monomials starting with the smallest.*

$$(0,0) <_{lex} (0,1) <_{lex} (0,2) \dots <_{lex} (1,0) <_{lex} (1,1) <_{lex} (1,2) \dots$$

$$1 <_{lex} x_2 <_{lex} x_2^2 \dots <_{lex} x_1 <_{lex} x_1 x_2 <_{lex} x_1 x_2^2 \dots$$

Maple has the command `TestOrder` *to test if a monomial is smaller than another one.*

```
with(Groebner);

TestOrder(x[1]*x[2],x[2]^5,plex(x[1],x[2]));
```

```
TestOrder(x[2]^4,x[1]^2*x[2],plex(x[1],x[2]));
```

The result of the first test will be false*, the second result will be* true*.*

For the next termorderings the total degree of monomials is needed. For a monomial $\underline{x}^{\alpha}$ the total degree is denoted by $|\alpha|$ where $|\alpha| = \alpha_1 + \cdots + \alpha_n$.

Definition 8. ***(Graded Lex Order)*** *Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ be elements of $\mathbb{Z}^n_{\geq 0}$. We say $\alpha >_{grlex} \beta$ if $|\alpha| > |\beta|$ or, in case of $|\alpha| = |\beta|$, if $\alpha >_{lex} \beta$.*

In this termorder monomials are first ordered by the total degree and then ties are broken using $>_{lex}$. The keyword for Maple is `grlex` and a possible termorder could be `grlex(x[2],x[1],x[3])`.

Example 21. *Here is a sketch how the monomials in $\mathbf{K}[x_1, x_2]$ are ordered in graded lex order.*

$$(0,0) <_{grlex} (0,1) <_{grlex} (1,0) <_{grlex} (0,2) <_{grlex} (1,1) <_{grlex} (2,0) \ldots$$

$$1 <_{grlex} x_2 <_{grlex} x_1 <_{grlex} x_2^2 <_{grlex} x_1 x_2 <_{grlex} x_1^2 \ldots$$

The corresponding Maple command is the same as in in Example 20, only `plex` *has to be replaced with* `grlex`*.*

Next the most commonly used termorder is discussed. Computations using this order tend to be faster than computations wrt. other orders.

Definition 9. ***(Graded Reverse Lex Order)*** *Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ be elements of $\mathbb{Z}^n_{\geq 0}$. We define $\alpha >_{grevlex} \beta$ if $|\alpha| > |\beta|$ or, in case of $|\alpha| = |\beta|$, if the rightmost nonzero entry of $\alpha - \beta \in \mathbb{Z}^n$ is negative.*

Again monomials are first ordered by the total degree. But ties are now broken using a different condition. In Maple this termorder is used with the keyword `tdeg`.

Example 22. *Here is a sketch how the monomials in $\mathbf{K}[x_1, x_2]$ are ordered when $<_{grevlex}$ is used.*

$$(0,0) <_{grevlex} (0,1) <_{grevlex} (1,0) <_{grevlex} (0,2) <_{grevlex} (1,1) <_{grevlex} (2,0) \ldots$$

$$1 <_{grevlex} x_2 <_{grevlex} x_1 <_{grevlex} x_2^2 <_{grevlex} x_1 x_2 <_{grevlex} x_1^2 \ldots$$

As it can easily be seen monomials in this example are sorted the same way as with the graded lex order. This is a special property of $>_{grevlex}$, namely if there are only two variables it leads to the same ordering as $>_{grlex}$. For three or more variables the ordering is really different, e.g.

$$x_1^2 x_3^2 >_{grlex} x_1 x_2^2 x_3 \quad \textit{but} \quad x_1^2 x_3^2 <_{grevlex} x_1 x_2^2 x_3.$$

All these orderings can also be combined which leads to the so called *Product Orders*. An example for a product order on $\mathbf{K}[x_1,x_2,x_3,x_4]$ in Maple would be `prod(plex(x[1],x[2]),tdeg(x[3],x[4]))`. This means that monomials are first compared using the `plex` order, ties are broken using the `tdeg` order. Even more than two partial orders are allowed.
It has to be noted explicitly that also the ordering of the variables can be varied, not only the type. All in all there are lots of different ways to define an order for ordering monomials.
With these preparations it is possible to order the monomials of a polynomial. Before the multivariate polynomial division can be defined another definition is necessary.

Definition 10. *Let $f \in \mathbf{K}[\underline{x}]$ be a polynomial with $f = \sum_\alpha a_\alpha x^\alpha$ and let $>_{\underline{x}}$ be a monomial ordering on $\mathbf{K}[\underline{x}]$. We define the **leading monomial** $LM(f)$ as the highest monomial of f with respect to $>_{\underline{x}}$, the **leading coefficient** $LC(f)$ as the coefficient of the highest monomial, and the **leading term** as $LT(f) = LC(f) \cdot LM(f)$.*

The most important issue in this definition is the leading monomial. It will appear quite often in the following definitions and theorems.

Example 23. *Let $f = x_1^2x_2^3 - 5x_1x_2x_3 + 4x_1^3x_3^2$ be a polynomial where the termorder $>_{grlex}$ on $\mathbf{K}[\underline{x}]$ is used:*

$$deg(f) = 5, \quad LM(f) = x_1^3x_3^2, \quad LC(f) = 4, \quad LT(f) = 4x_1^3x_3^2.$$

To obtain these results with Maple the following commands can be used:

```
with(Groebner);

f:=x[1]^2*x[2]^3-5*x[1]*x[2]*x[3]+4*x[1]^3*x[3]^2;

degree(f,[x[1],x[2],x[3]]);

LeadingMonomial(f,grlex(x[1],x[2],x[3]));

LeadingCoefficient(f,grlex(x[1],x[2],x[3]));

LeadingTerm(f,grlex(x[1],x[2],x[3]));
```

For the leading term Maple does not return the product, but the pair $LC(f), LM(f)$. If the termorder `plex(x[2],x[3],x[1])` *is used instead the results are as follows:*

$$deg(f) = 5, \quad LM(f) = x_1^2x_2^3, \quad LC(f) = 1, \quad LT(f) = x_1^2x_2^3.$$

Now all the ingredients are defined and it is possible to introduce the multivariate division.

Definition 11. *Let $F = [f_1, \ldots, f_s]$ be an ordered list of polynomials in $\mathbf{K}[\underline{x}]$ and $>_{\underline{x}}$ a monomial order. Then every polynomial $f \in \mathbf{K}[\underline{x}]$ can be written in the form*

$$f = a_1 f_1 + \cdots + a_s f_s + r$$

where all a_i and r are elements of $\mathbf{K}[\underline{x}]$ and r is either 0 or a polynomial where no monomial is divisible by any of $LM(f_1), \ldots, LM(f_s)$. r is called ***a remainder*** *of f on division by F.*

The algorithm which produces the a_i and the remainder r acts in the following way: It is tested if $LM(f)$ is divisible by $LM(f_1)$, then by $LM(f_2)$ and so on. If the result of such a test is true for a leading monomial $LM(f_i)$, testing is stopped and an appropriate multiple of f_i is subtracted from f such that $LT(f)$ is cancelled. The result is again named f and the testing starts again. In the other case that $LM(f)$ is not divisible by $LM(f_1), \ldots, LM(f_s)$, the term $LT(f)$ is added to the remainder. It follows that in each step $LT(f)$ is removed, either by cancellation or by moving it to the remainder. The process is finished when f is 0.

Example 24. *As an example the polynomial $f = x_1^2 x_2 + x_1 x_2^2 + x_2^2$ is divided by the polynomials $f_1 = x_1 x_2 - 1$ and $f_2 = x_2^2 - 1$, where the termorder $>_{lex}$ is used in $\mathbf{K}[x_1, x_2]$.*

$$\begin{aligned}
&f = x_1^2 x_2 + x_1 x_2^2 + x_2^2, \quad r = 0,\ a_1 = 0,\ a_2 = 0,\\
&f \leftarrow x_1 x_2^2 + x_1 + x_2^2, \quad r = 0,\ a_1 = x_1,\ a_2 = 0,\\
&f \leftarrow x_1 + x_2^2 + x_2, \quad r = 0,\ a_1 = x_1 + x_2,\ a_2 = 0,\\
&f \leftarrow x_2^2 + x_2, \quad r = x_1,\ a_1 = x_1 + x_2,\ a_2 = 0,\\
&f \leftarrow x_2 + 1, \quad r = x_1,\ a_1 = x_1 + x_2,\ a_2 = 1,\\
&f \leftarrow 1, \quad r = x_1 + x_2,\ a_1 = x_1 + x_2,\ a_2 = 1,\\
&f \leftarrow 0, \quad r = x_1 + x_2 + 1,\ a_1 = x_1 + x_2,\ a_2 = 1.
\end{aligned}$$

It follows that f can be written as

$$f = a_1 f_1 + a_2 f_2 + r = (x_1 + x_2)(x_1 x_2 - 1) + (1)(x_2^2 - 1) + (x_1 + x_2 + 1),$$

where no monomial in the remainder $x_1 + x_2 + 1$ is divisible by $LM(f_1)$ or $LM(f_2)$.

It has to be said clearly that the result r is *a remainder* of f on division by the ordered list F. If f_1 is exchanged with f_2 the following result is obtained:

$$f = a_1' f_1 + a_2' f_2 + r' = (x_1 + 1)(x_2^2 - 1) + (x_1)(x_1 x_2 - 1) + (2x_1 + 1).$$

So the remainder r depends on the order of the polynomials in the list F and of course, on the monomial order which has to be chosen first of all.

Using the multivariate division the notion of *reduction* can be defined.

Definition 12. *Let $F = [f_1, \ldots, f_s]$ be an ordered list of polynomials in $\mathbf{K}[\underline{x}]$, $f \in \mathbf{K}[\underline{x}]$ and $>_{\underline{x}}$ a monomial order. Then we call the process of dividing f by F **reduction** and denote the remainder with $\overline{f}^F$. The choice of a monomial order is required to be able to compute the remainder.*

This reduction will be used quite often in the following definitions and theorems. But before that the corresponding Maple command is given.

Example 25. *The polynomial $f = x_1^2x_2 + x_1x_2^2 + x_2^2$ is to be divided by the polynomials $f_1 = x_1x_2 - 1$ and $f_2 = x_2^2 - 1$, where the termorder $>_{lex}$ is used in $\mathbf{K}[x_1, x_2]$.*

```
with(Groebner);

f:=x[1]^2*x[2]+x[1]*x[2]^2+x[2]^2;

f1:=x[1]*x[2]-1;

f2:=x[2]^2-1;

r:=Reduce(f,[f1,f2],plex(x[1],x[2]),'s','a');

s;  a;
```

The result is a remainder r, a list of quotients a and a number s such that

$$f = \sum_{i=1}^{2} a_i f_i + \frac{r}{s}.$$

The result for this example is:

$$r = x_1 + x_2 + 1, \qquad a = [x_1 + x_2, 1,] \qquad s = 1.$$

The necessity of a monomial order will not be mentioned explicitly from now on. Next we define *interreduction*.

Definition 13. *Let $F = [f_1, \ldots, f_s]$ be an ordered list of polynomials in $\mathbf{K}[\underline{x}]$. The process of replacing each polynomial f_i by $\overline{f_i}^{F\setminus\{f_i\}}$ is called **interreduction** of the list F.*

This means that every polynomial in F is reduced with respect to all other elements of the list. An important property of an interreduction is that the original set of polynomials and the result of the interreduction generate the same ideal. Interreduction can sometimes be used to simplify generating sets (bases) of an ideal, to obtain shorter polynomials. The right choice of the monomial order is important.

Example 26. *The ideal* $I = \langle (x_1-1)^2+(x_2+5)^2-6, 2x_1^2+2x_2^2-4\rangle$ *is interreduced. The appropriate commands are the following:*

```
with(Groebner);

I1:=[(x[1]-1)^2+(x[2]+5)^2-6,2*x[1]^2+2*x[2]^2-5];

ir:=InterReduce(I1,plex(x[1],x[2]));
```

And the result is

$$ir = [416x_2^2+1800x_2+1985, 4x_1-20x_2-45].$$

Example 27. *Interreduction is very efficient for the constraint polynomials* h_1, h_2, h_3 *of Example3:*

```
Groebner[InterReduce]([h_1,h_2,h_3],tdeg(y_1,y_2,x_3));
```

yields the three simplified polynomials

$$\begin{aligned}
g_1 :\ & (-5239\sqrt{3}-1876)x_3^2+(624\sqrt{3}y_2+4550\sqrt{3}+174y_2+11684)x_3+104\sqrt{3}y_1 \\
& -1755\sqrt{3}+218y_1-720y_2-774=0, \\
g_2 :\ & 547x_3^2+(18y_1-48y_2-476)x_3-8y_1+18y_2+99=0, \\
g_3 :\ & -8527x_3^2+(552y_2+7616)x_3+36y_1^2+36y_2^2+344y_1-1359=0,
\end{aligned}$$

which have the property that y_1 *and* y_2 *appear linear.*

Now back to ideals generated by a set of polynomials. There was already the question how one could decide if a given polynomial f is an element of $I = \langle f_1, \ldots, f_s\rangle$. The concept of reduction would be a good method to answer that question. First a result is given which is not the perfect solution.

Theorem 2. *Let* $I = \langle f_1, \ldots, f_s\rangle \subseteq \mathbf{K}[\underline{x}]$ *be an ideal and* $f \in \mathbf{K}[\underline{x}]$. *If* $\overline{f}^F = 0$ *then* f *is an element of* I.

The disadvantage of this result is that if $\overline{f}^F \neq 0$ the question is open as before. The crucial point is that the leading terms $LT(f_1),\ldots,LT(f_s)$ which are used in the division algorithm are in general "bad" representatives for the set of all leading terms which are possible in the ideal I.

Example 28. *Let $F = [f_1, f_2] = [x_1x_2 + 1, x_2^2 - 1]$ be an ordered list of polynomials in $\mathbf{K}[\underline{x}]$ and $p_1 = x_1f_1 + x_2f_2$ and $p_2 = x_2f_1 + x_1f_2$ be combinations of these polynomials. It is clear that $p_1, p_2 \in \langle f_1, f_2\rangle$. But if the reductions are computed wrt.* `plex(x[1],x[2])`*, the following results are obtained:*

$$\overline{p_1}^F = 0, \qquad \overline{p_2}^F = -x_1 - x_2.$$

In the first case Theorem 2 works quite fine, it does not in the second.

Now *standard bases* are defined which are "much better" representatives (generators) for an ideal. They are still dependent from a chosen monomial order but nevertheless very useful to deduce information about the ideal and the corresponding variety.

4.2 Standard Bases

Definition 14. *For a fixed monomial order and an ideal $I \in \mathbf{K}[\underline{x}]$ a finite subset $G = \{g_1,\ldots,g_t\}$ of I is called a **Groebner basis** or **standard basis** if*

$$\langle LM(g_1),\ldots,LM(g_t)\rangle = \langle LM(I)\rangle,$$

where $LM(I)$ is the ideal generated by all the leading terms of the elements of I.

With other words a basis is also a Groebner basis if the leading monomials of the generators generate the same ideal as the leading monomials of all ideal elements.
A Groebner basis is a very special generating set of the ideal I with some nice properties. Such a basis is not unique due to the fact that adding another polynomial to the generators does not change this property. Another reason for non-uniqueness is the fact that a monomial order has to be chosen first.
In this introduction it will not be explained in detail how such a Groebner basis is computed explicitly, starting with a set of generators and a term order. Just a short appetizer: For all possible pairs of generators the so called *S-polynomials* are computed and reduced with the list of generators. All remainders which are not 0 are added to the list and the procedure starts again. Such rounds are repeated until all remainders of the S-polynomials are all 0 after reduction. See the book "Ideals, Varieties, and Algorithms" by Cox, Little and O'Shea for a detailed description of the algorithm and enhancements of it.
With a Groebner basis we get a better result for the ideal-membership-question.

Theorem 3. *Let $G = \{g_1, \ldots, g_t\} \subseteq \mathbf{K}[\underline{x}]$ be a Groebner basis of an ideal I and $f \in \mathbf{K}[\underline{x}]$. The polynomial f is an element of I if and only if $\overline{f}^G = 0$.*

All in all the procedure for the ideal-membership-question is the following: first a termorder is fixed, then a Groebner basis of the ideal is computed and after that the polynomial f is reduced with respect to this basis. f is only then an element of I if the result of the reduction is 0.
It has to be mentioned that the result of a reduction with respect to a Groebner basis is independent of the order of the basis elements. This was not the case when the reduction was done with respect to a normal basis.

Example 29. *As an example we take the polynomials from Example 28. Let $I = \langle f_1, f_2 \rangle = \langle x_1x_2 + 1, x_2^2 - 1 \rangle$ be an ideal in $\mathbf{K}[\underline{x}]$ and $p_1 = x_1f_1 + x_2f_2$ and $p_2 = x_2f_1 + x_1f_2$ polynomials which are definitely elements of I.*
Now a Groebner basis is computed wrt. `plex(x[1],x[2])` *and p_1, p_2 are reduced with that basis.*

```
with(Groebner);

with(PolynomialIdeals):

f1:=x[1]*x[2]+1;  f2:=x[2]^2-1;

G:=Basis(<f1,f2>,plex(x[1],x[2]));

p1:=x[1]*f1+x[2]*f2;

p2:=x[2]*f1+x[1]*f2;

Reduce(p1,G,plex(x[1],x[2]));

Reduce(p2,G,plex(x[1],x[2]));
```

The Groebner basis is

$$G = \langle x_2^2 - 1, x_1 + x_2 \rangle$$

and now the expected results are obtained:

$$\overline{p_1}^F = 0, \qquad \overline{p_2}^F = 0.$$

The package `PolynomialIdeals` *contains also two commands for testing if a polynomial or an ideal is contained in another ideal. They are used as follows:*

```
IdealMembership(p1,G);

IdealContainment(<p1,p2>,G);
```

The result of both commands will be true *for this example.*

As mentioned above adding another polynomial of the ideal to the generators does not change the property of being a Groebner basis. To get a unique version of such a basis the so called *reduced Groebner bases* are defined.

Definition 15. *Let $G = \{g_1, \ldots, g_t\} \subseteq \mathbf{K}[x]$ be a Groebner basis of a given ideal I. The generators G are a **reduced Groebner basis** if G is interreduced and all leading coefficients are 1.*

Fortunately all bases returned by Maple are interreduced. So at least the more important condition is fulfilled and the bases are quasi unique. This is not the case when *Singular* is used. But there exists a command to switch on basis-interreduction.
Now some remarks about Groebner bases:

- The number of polynomials in a Groebner basis can be very large, it depends on the complexity of the ideal and, of course, on the chosen monomial order. It is quite possible that only a part of the Groebner basis would be enough to generate the ideal I, but then the relevant condition $\langle LM(g_1), \ldots, LM(g_t)\rangle = \langle LM(I)\rangle$ is no more fulfilled.
- There are different methods to compute Groebner bases. The sketch of the algorithm above describes only the main concept. Lots of improvements were made to accelerate basis computations. Depending on the termorder the computations can be very expensive with respect to time and memory. Sometimes it is better to compute the basis for an "easy" termorder and then to convert it to the desired termorder using one of the conversion-algorithms.
- The monomial order $>_{lex}$ (in Maple `plex`) tends to be expensive in general, whereas the order $>_{grevlex}$ (in Maple `tdeg`) tends to be "relatively" cheap. If the type of order is chosen the computational costs can again be influenced by the order of unknowns.
- The typical way to order the polynomials in a basis is by sorting them wrt. the leading monomials, starting with the smallest one.

Example 30. *Now as an example reduced Groebner bases are computed for the ideal*

$$I = \langle x_1^2 x_2 + x_1 x_2^2 + x_2^2 x_3, x_1 x_2 - x_3, x_2^2 - 1\rangle$$

using different termorders:

$>_{lex}$ *in* $\mathbf{K}[x_1,x_2,x_3]$:

$$G = \{x_3^3+2x_3^2, x_2x_3+x_3^2+x_3, x_2^2-1, x_1+x_3^2+x_3\}$$

$>_{lex}$ *in* $\mathbf{K}[x_3,x_1,x_2]$:

$$G = \{x_2^2-1, x_1^2+x_1x_2+x_1, x_3-x_1x_2\}$$

$>_{grlex}$ *in* $\mathbf{K}[x_1,x_2,x_3]$:

$$G = \{x_3^2+x_1+x_3, x_2x_3-x_1, x_2^2-1, x_1x_3+x_1+x_3, x_1x_2-x_3, x_1^2+x_1+x_3\}$$

The code for Maple looks as follows:

```
with(Groebner);

with(PolynomialIdeals):

f1:=x[1]^2*x[2]+x[1]*x[2]^2+x[2]^2*x[3];

f2:=x[1]*x[2]-x[3];  f3:=x[2]^2-1;

Basis(<f1,f2,f3>,plex(x[1],x[2],x[3]));

Basis(<f1,f2,f3>,plex(x[3],x[1],x[2]));

Basis(<f1,f2,f3>,grlex(x[1],x[2],x[3]));
```

It is no coincidence that in all the bases wrt. to $>_{lex}$ the first polynomial is univariate. This is because the lex-orders have the so called *elimination property*.

Example 31. *Taking the three polynomial equations g_1,g_2,g_3 from Example 27 and computing a Groebner basis using the command*

```
Groebner[Basis]([g_1,g_2,g_3],plex(y1,y2,x3));
```

yields immediately a univariate polynomial of degree six in x_3 and two more polynomials in which y_1 and y_2 appear linearly.

What is elimination resp. an elimination ideal?

4.3 Elimination

First of all the so called *elimination ideals* are defined.

Definition 16. *Let I be an ideal in $\mathbf{K}[\underline{x}] = \mathbf{K}[x_1,\ldots,x_n]$ and $1 \leq l < n$. Then the ideal $I_l = I \cap \mathbf{K}[x_{l+1},\ldots,x_n]$ is called the **l-th elimination ideal** of I. I_l contains all elements of I which do not contain the variables $x_1,\ldots,x_l$.*

If a termorder has the elimination property, then a basis wrt. this order can be used to extract bases for the elimination ideals.

Theorem 4. *Let $G = \{g_1,\ldots,g_t\} \subseteq \mathbf{K}[x_1,\ldots,x_n]$ be a Groebner basis of an ideal I with respect to $>_{lex}$. Furthermore let $H = \{h_1,\ldots,h_k\}$ be the first k polynomials of G which do not contain the unknowns $x_1,\ldots,x_l$. Then H is a Groebner basis of the l-th elimination ideal I_l.*

It follows an example where it can be seen more clearly what this means.

Example 32. *The ideal $I = \langle x_1^2x_2 + x_1x_2^2 + x_2^2x_3, x_1x_2 - x_3, x_2^2 - 1\rangle$ is given with the Groebner basis wrt.* `plex(x[1],x[2],x[3])`

$$G = \langle x_3^3 + 2x_3^2, x_2x_3 + x_3^2 + x_3, x_2^2 - 1, x_1 + x_3^2 + x_3\rangle.$$

Then the following generators are all Groebner bases of the corresponding elimination ideal:

$$\begin{aligned} I_2 &= \langle x_3^3 + 2x_3^2\rangle, \\ I_1 &= \langle x_3^3 + 2x_3^2, x_2x_3 + x_3^2 + x_3, x_2^2 - 1\rangle. \end{aligned}$$

There is also another command to do the elimination directly. It is contained in the package `PolynomialIdeals` *and to compute I_1 directly from the input polynomials it is used as follows:*

```
EliminationIdeal(<f1,f2,f3>,{x[2],x[3]});
```

The second argument gives the unknowns which shall not be eliminated. Furthermore no termorder has to be chosen, this is done by Maple internally. Unfortunately this choice is session-dependent.

It can easily be seen that with these elimination ideals it is quite easy to solve a system of equations with only finitely many solutions.

Example 33. *Here once again the Groebner basis G from above including I_2 and I_1.*

$$\begin{aligned} I_2 &= \langle x_3^3 + 2x_3^2\rangle, \\ I_1 &= \langle x_3^3 + 2x_3^2, x_2x_3 + x_3^2 + x_3, x_2^2 - 1\rangle, \\ G &= \langle x_3^3 + 2x_3^2, x_2x_3 + x_3^2 + x_3, x_2^2 - 1, x_1 + x_3^2 + x_3\rangle. \end{aligned}$$

Now the system can be solved step by step by solving the partial systems and extending the solutions.

```
with(Groebner);

with(PolynomialIdeals):

f1:=x[1]^2*x[2]+x[1]*x[2]^2+x[2]^2*x[3];

f2:=x[1]*x[2]-x[3];  f3:=x[2]^2-1;

I0:=<f1,f2,f3>;

G:=<op(Basis(I0,plex(x[1],x[2],x[3])))>;

NumberOfSolutions(I0);

I2:=EliminationIdeal(<f1,f2,f3>,{x[3]});

I1:=EliminationIdeal(<f1,f2,f3>,{x[2],x[3]});

L1:={solve(Generators(I2),{x[3]})};

L2:=map(y->op(map(z->z union y,{solve(eval(Generators(I1),y),
                                      {x[2]})})),L1);

L3:=map(y->op(map(z->z union y,{solve(eval(Generators(G),y),
                                      {x[1]})})),L2);
```

The result for the vanishing set is $\mathbf{V}(I) = \{(-2,1,-2),(0,-1,0),(0,1,0)\}$*, where the second solution has multiplicity 2.*

How does the variety of the l-th elimination ideal $\mathbf{V}(I_l)$ correspond to the original variety $\mathbf{V}(I)$?

Theorem 5. *The variety* $\mathbf{V}(I_l) \in \mathbf{K}^{n-l}$ *is the smallest variety (wrt. inclusion) which contains* $\pi(\mathbf{V}(I))$*, where* $\pi(\mathbf{V}(I))$ *is the orthonormal projection of* $\mathbf{V}(I)$ *onto the subspace generated by equations* $x_1 = \cdots = x_l = 0$*.*

An important fact is that the equality $\mathbf{V}(I_l) = \pi(\mathbf{V}(I))$ only holds when I is a *projective variety* and I_l a *projective elimination ideal*. This means that here it can happen that a solution of I_l can not be extended to a solution of I_{l-1}.

Another application of elimination is implicitization. The following example shows how a variety can be deduced from a parametrisation.

Example 34. *A parametrisation of an ellipse is given.*

$$x[1] = \frac{5(1-t^2)}{1+t^2} \qquad x[2] = \frac{3(2t)}{1+t^2}$$

For all values of $t \in \mathbb{R}$ a point of the ellipse is obtained. The only point which cannot be reached is the left apex $(-5,0)$. To compute the smallest variety which contains all these points (should be the ellipse), the following code could be used:

```
with(Groebner);


with(PolynomialIdeals):

par1:=x[1]-(5*(1-t^2))/(1+t^2);

par2:=x[2]-(3*(2*t))/(1+t^2);

J:=<numer(par1),numer(par2)>;

G:=Basis(J,plex(t,x[1],x[2]));

J1:=<G[1]>;
```

The first elimination ideal $J_1 = \langle 9x_1^2 + 25x_2^2 - 225 \rangle$ has exactly the ellipse as vanishing set.

In the last subsection two very important concepts are discussed that play major roles in the analysis of more complicated kinematic structures. The first one has to do with the degree of freedom of a mechanism and the second one can help to find different operation modes of a kinematic structure.

4.4 Dimension, Primary Decomposition

It happens quite often that one is first of all interested in the *dimension* of a variety. Here is an example where the variety contains parts with different dimensions and it is shown how the dimension is computed.

Example 35. *The ideal* $I = \langle (2x_1 - x_2 - 2)x_1, (2x_1 - x_2 - 2)x_2^2 \rangle$ *is given.* $\boldsymbol{V}(I)$ *consists of a line and an isolated point. Using the Hilbert polynomial the dimension is computed (which should be 1).*

```
with(PolynomialIdeals):

I1:=<(2*x[1]-x[2]-2)*x[1], (2*x[1]-x[2]-2)*x[2]^2>;

HilbertDimension(I1,{x[1],x[2]});
```

If a variety contains parts with different dimensions, then the dimension of the whole variety is defined to be the largest of these numbers.

In kinematics the dimension of the ideal generated by the constraint equations yields the degrees of freedom of the mechanical structure. But one has to be very careful with the obtained number. As mentioned above the Hilbert Dimension yields the largest dimension. At special points of the corresponding variety this number can be larger or smaller.

Example 36. *Taking the three constraint equations* h_1, h_2, h_3 *of Example 3 and computing the Hilbert dimension of the generated ideal one can use the Maple command*

```
Groebner[HilbertDimension]([h1,h2,h3],tdeg(x3,y1,y2));
```

The answer will be 0 *and this means that the solution of the corresponding system of equations is a set of points. The command*

```
Groebner[Basis]([h1,h2,h3],plex(y1,y2,x3));
```

yields immediately, without further manipulations, a univariate polynomial of degree 6 in x_3 *and two more polynomials which are linear in* y_1 *and* y_2.

Substituting on the other hand the following set of design and input parameters

$$A_2 = 16, A_3 = 9, B_3 = 6, a_2 = 16, a_3 = 9, b_3 = 6, l_1 = 10, l_2 = 10, l_3 = 10,$$
$$k_1 = 5, k_2 = 5, k_3 = 5, u = 1, v = 1, w = 1,$$

into the general system of constraint equations Eq.(20) *one obtains a new system of constraint equation which reads:*

$$g_1 : 75x_3^2 + 40x_3y_1 + 4y_1^2 + 4y_2^2 + 40y_2 + 75 = 0,$$
$$g_2 : 1099x_3^2 + 40x_3y_1 - 128x_3y_2 + 4y_1^2 + 4y_2^2 - 640x_3 + 40y_2 + 75 = 0,$$
$$g_3 : 783x_3^2 + 88x_3y1 - 72x_3y_2 + 4y_1^2 + 4y_2^2 - 360x_3 + 40y_2 + 75 = 0.$$

Invoking again the Maple command

```
Groebner[HilbertDimension]([g1,g2,g3],tdeg(x3,y1,y2));
```

yields now the answer 1 for the Hilbert dimension. This means that the manipulator allows a one dof motion with all inputs locked. In this case the inputs were all set to 1, which corresponds to 90°*. The Groebner base for this set of equations is:*

```
Groebner[Basis]([h1,h2,h3],plex(y1,y2,x3));
[x3*(201*x3^2-20), -x3*(8*x3-y2-5), x3*(11*x3+4*y1),
-35*x3^2+4*y1^2+4*y2^2+40*y2+75].
```

The base consists of four equations and the first equation is $x_3(201x_3^2 - 20) = 0$*. One can see immediately that the solution* $x_3 = 0$ *causes the second and the third equation to vanish. Only the fourth equation remains.* $x_3 = 0$ *and* $4y_1^2 + 4y_2^2 + 40y_2 + 75 = 0$ *determine the self motion, which is the well known parallel bar motion. But when* $x_3 \neq 0$ *then the four equations have two distinct solutions, one of them reads*

$$x_3 = \frac{2}{201}\sqrt{1005},\ y_1 = -\frac{11}{402}\sqrt{1005},\ y_2 = \frac{16}{201}\sqrt{1005} - 5.$$

The two solutions yield two distinct assemblies of the manipulators. From point of view of algebraic geometry the ideal has two components which have different dimensions. The corresponding affine varieties consist of a curve (Fig.18, left) and two points (Fig.18, right). In this example it is not possible to to move directly from one component to the other. One has to disassemble the mechanism and to reassemble to obtain a solution of the other component. Each component corresponds to an operation mode of the mechanism or robot. As it will be shown in the next section it can happen that different components can have intersections and this would mean that a mechanism is able to to move at certain poses from one operation mode into an other operation mode.

As already mentioned it is possible that an ideal describes a variety which is made up of some simpler varieties, e.g. the variety from the Examples 35 and 36. The question is how such a decomposition can be found, if there is one.

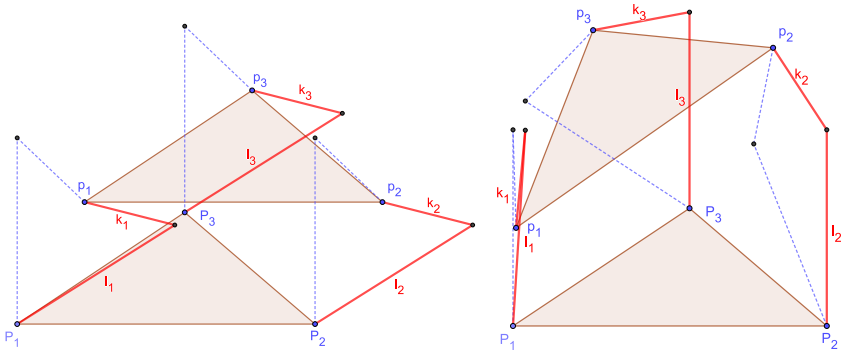


Figure 18. Different Hilbert dimensions of the constraint equations g_1, g_2, g_3 left: self motion, right: rigid assembly

Definition 17. *A **primary decomposition** of a given ideal I is an expression of I as an intersection of primary ideals, namely $I = \bigcap_{i=1}^{r} Q_i$. Such a decomposition is called **minimal** if the radicals $\sqrt{Q_i}$ are all different and $Q_i \not\supseteq \bigcap_{i \neq j} Q_j$. Furthermore if no radical $\sqrt{Q_i}$ is strictly contained in another radical $\sqrt{Q_j}$, then the primary components Q_i are uniquely determined.*
The radicals $\sqrt{Q_i} =: P_i$ are the corresponding prime ideals.

It follows an example where the variety can be decomposed into three different parts.

Example 37. *A rather large ideal J is given.*

$$\begin{aligned} J \;=\; & \langle 4x_3{}^2x_1{}^3 + 4x_3{}^2x_1x_2{}^2 + 4x_3{}^4x_1 - 16x_1x_3{}^2 + 5x_3{}^2x_2x_1{}^2 + 5x_3{}^2x_2{}^3 + \\ & + 5x_3{}^4x_2 - 20x_2x_3{}^2 - 6x_3{}^2x_1{}^2 - 6x_3{}^2x_2{}^2 - 6x_3{}^4 + 24x_3{}^2, \\ & 4x_1{}^4 + 4x_1{}^2x_2{}^2 + 4x_3{}^2x_1{}^2 + 2x_1{}^2 + 5x_1{}^3x_2 + 5x_1x_2{}^3 + 5x_1x_2x_3{}^2 - \\ & - 20x_1x_2 - 18x_1{}^3 - 18x_1x_2{}^2 - 18x_1x_3{}^2 + 72x_1 - 15x_2x_1{}^2 - 15x_2{}^3 - \\ & - 15x_2x_3{}^2 + 60x_2 + 18x_2{}^2 + 18x_3{}^2 - 72\rangle \end{aligned}$$

Using the command

```
PrimaryDecomposition(J);
```

the primary decomposition is computed and the result is

$$Q_1 = \langle 4x_1 + 5x_2 - 6\rangle, \qquad Q_2 = \langle x_1{}^2 + x_2{}^2 + x_3{}^2 - 4\rangle, \qquad Q_3 = \langle x_3{}^2, x_1 - 3\rangle,$$

which means that $\mathbf{V}(I)$ can be decomposed into a line, a sphere and an isolated point which appears with multiplicity 2.

Such decompositions are quite convenient if the variety has to be intersected with other varieties, because then each of the components can be treated separately. The worst case is that one has to deal with an ideal which is primary or even prime. In this case one has to take the ideal as a whole.

Examples of the application of primary decomposition in the analysis of parallel manipulators are given in Section 5.

As a final comment is has to be mentioned that it is possible to extend the concept of ideals and varieties to projective spaces. Due to the fact that lots of subtleties appear there, the reader is once more referred to the book *"Ideals, Varieties, and Algorithms"* by *David Cox, John Little* and *Donal O'Shea*.

5 Complete kinematic analysis of the TSAI parallel manipulator

Within this section the algebraic methods derived in the previous sections are applied to the complete kinematic analysis of a parallel manipulator introduced by L.W. Tsai in 1996. It will be shown how the direct kinematics can be solved. Moreover this manipulator exhibits different interesting operation modes, which can be detected using primary decomposition. The analysis continues with a singularity analysis of each operation mode and the discussion of all poses in which the manipulator can change from one operation mode into another.

5.1 Introduction

In 1996 L.W. Tsai (Tsai (1996)) designed 3-DOF parallel manipulators with the aim to generate pure translational motion. As a result he presented 3-UPU parallel manipulators where the axes of the U-joints have to fulfill some special conditions in order to allow the mechanism a translational motion. A generalization followed in 1998 by Di Gregorio and Parenti-Castelli by decomposing the U-joints into two simple R-joints without any condition of intersection or perpendicularity. There exist many publications about singularities of these mechanisms, especially because the SNU 3-UPU, which was built in about 2001 at the Seoul National University, showed unexpected motion in its home position, leading to many different attempts to explain this behaviour, see e.g. Bonev and Zlatanov (2001), Wolf et al. (2002), Han et al. (2002). A complete analysis using algebraic methods can be found in Walter et al. (2009), where also the strange behaviour of the mechanism at the home pose is explained. The crucial point in the design of the SNU 3-UPU is the fact that all axes in the base (resp. platform) intersect in a single point. The type of manipulator which will be discussed in detail in this section is the classic 3-UPU mechanism, where the axes of the U-joints are in a more general position. An analysis of the manipulator's translational kinematics was published

by Di Gregorio and Parenti-Castelli (1999) where also singular configurations are discussed. Concerning such configurations there is also an interesting thesis by Chebbi (2011), where 3-UPU mechanisms are discussed in detail regarding singularities and strategies for selecting appropriate architectures (see also Chebbi and Parenti-Castelli (2010)). The topic of workspace optimization is covered by several publications, e.g. Parenti-Castelli et al. (2000), Badescu et al. (2002), Tsai and Joshi (2000), and also Dwarakanath and Bhutani (2012) where the capabilities of the translational 3-UPU manipulator are tested with a high precision prototype model. Another part of publications about 3-UPU mechanisms discusses generalizations and slightly exotic varieties of the manipulator, e.g. Ji and Wu (2003) and Varedi et al. (2009) where the U-joints are decomposed into R-joints with offets, and Hu and Lu (2011) where the first axes of the lower U-joints are perpendicular to the base.

The topic of this section is the classical fully symmetric TSAI 3-UPU mechanism, where the anchor points in the base (resp. platform) constitute an equilateral triangle, and the axes embedded in base and platform are tangent to the circumcircle of the corresponding triangle. Goal of this section is to analyze the motion capabilities of this manipulator by a purely algebraic approach, which was developed in the previous sections of this chapter in terms of a description by algebraic constraint equations. Due to the fact that the manipulator's translational motion was the main issue in most publications, the question about other types of motion that are allowed by this design, arises immediately. The present description via algebraic equations gives a complete anlysis about absolutely all poses the mechanism can attain and allows therefore a more general view. Some results of other authors appear again naturally as a part of this analysis, e.g. the singular poses of the translational operation mode.

5.2 Manipulator design

First of all an exact description of the manipulator's design is necessary. Figure 19 depicts a sketch of the manipulator including all necessary notations. Due to the fact that the TSAI 3-UPU can be obtained from the SNU 3-UPU by simply rotating each limb by 90 degrees about the axis of the prismatic joint, the following is almost identical to the description of the SNU 3-UPU in Walter et al. (2009). In the base there are three points $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ forming an equilateral triangle with circumradius h_1. The frame Σ_0 is fixed in the base such that its origin lies in the circumcenter of the triangle, its yz-plane coincides with the plane containing the triangle and its z-axis passes through $\mathbf{A}_3$. The same situation is established in the platform. There we have an equilateral triangle with vertices $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$, circumradius h_2, and the platform frame is denoted by Σ_1. The parameters h_1 and h_2 are the first two design parameters, which are always different, if not mentioned

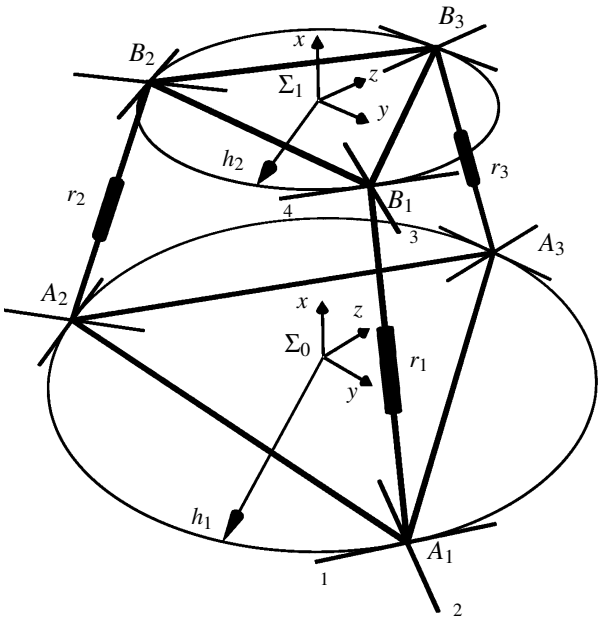


Figure 19. TSAI 3-UPU manipulator

explicitly. Each pair of corresponding points $\mathbf{A}_i$, $\mathbf{B}_i$ is connected by a limb L_i with U-joints at each end. The length of each limb is denoted by r_i and adjusted by a prismatic joint. The first and the fourth axis are embedded in the base resp. platform such that each of them is tangent to the corresponding circumcircle. Second and third axis of this link-combination are parallel to each other and perpendicular to the axis of the limb and its first and fourth axis. With this setup the conditions for translational motion as stated in Di Gregorio and Parenti-Castelli (1998) are fulfilled. All together there are five parameters to describe the TSAI 3-UPU mechanism: r_1, r_2, r_3, h_1, and h_2, where the first three of these are used to actuate the manipulator, the latter are fixed design parameters.

5.3 Deduction of constraint equations

In the following algebraic equations are derived whose solutions describe the pose of Σ_1 with respect to Σ_0, and with it the pose of the platform with respect to the fixed base. For each limb L_i the kinematic chain can be described by a sequence of transformations which was already explained in Eq.(40) on page 26. The pose of frame Σ_1 with respect to Σ_0 via limb L_i can be described by the matrix

product

$$\mathbf{T}_i = \mathbf{F}_{i1}\mathbf{M}_{i1}\mathbf{G}_{i1}\mathbf{M}_{i2}\mathbf{G}_{i2}\mathbf{M}_{i3}\mathbf{G}_{i3}\mathbf{M}_{i4}\mathbf{F}_{i2}. \tag{57}$$

The matrices $\mathbf{F}_{ij}$ at the beginning and at the end are fixed transformations, only depending on the design parameters h_1 and h_2 and the location of the anchor points:

$$\mathbf{F}_{11} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{3}}{2}h_1 & 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2}h_1 & 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}, \quad \mathbf{F}_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -h_2 & 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix},$$

$$\mathbf{F}_{21} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{3}}{2}h_1 & 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ -\frac{1}{2}h_1 & 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}, \quad \mathbf{F}_{22} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ h_2 & 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix},$$

$$\mathbf{F}_{31} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ h_1 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{F}_{32} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -h_2 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

Matrices $\mathbf{M}_{ij}$ are responsible for the passive rotations about the axes of the U-joints depending on rotation angles u_{ij}, and the matrices $\mathbf{G_{ij}}$ manage the transformations from one rotational axis to the next one:

$$\mathbf{M}_{ij} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(u_{ij}) & -\sin(u_{ij}) & 0 \\ 0 & \sin(u_{ij}) & \cos(u_{ij}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{G}_{ij} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a_j & 1 & 0 & 0 \\ 0 & 0 & \cos(\alpha_j) & -\sin(\alpha_j) \\ d_j & 0 & \sin(\alpha_j) & \cos(\alpha_j) \end{pmatrix}.$$

The corresponding Denavit-Hartenberg parameters are given in Table 3.

Table 2. Denavit-Hartenberg parameters for limb L_i

j	a_j	d_j	α_j
1	0	0	$\pi/2$
2	r_i	0	0
3	0	0	$-\pi/2$

Due to the special arrangement of the axes in each limb no separate transformation matrix was introduced for the active prismatic joints, their parameter r_i appears in the matrix $\mathbf{G}_{i2}$. A clear and brief introduction of the Denavit-Hartenberg

convention and the systematic deduction of the forward transformation matrices $\mathbf{T}_i$ can be found e.g. in Tsai (1999), Husty et al. (1997), and Pfurner (2006).

In a next step half-tangent substitutions for all u_{ij} are performed to get rid of the trigonometric functions:

$$\cos(u_{ij}) = \frac{1-t_{ij}^2}{1+t_{ij}^2}, \quad \sin(u_{ij}) = \frac{2t_{ij}}{1+t_{ij}^2} \tag{58}$$

This results in displacement matrices $\overline{\mathbf{T}}_i$ containing new parameters t_{ij} for the rotation angles of the passive joints. Then from each matrix $\overline{\mathbf{T}}_i$ the Study parameters are computed using Eq.(7) and (8), leading to expressions

$$\begin{aligned} x_0 &= f_{i1}(t_{i1},t_{i2},t_{i3},t_{i4}) \\ &\vdots \\ y_3 &= f_{8i}(t_{i1},t_{i2},t_{i3},t_{i4}) \quad i=1,\ldots,3. \end{aligned} \tag{59}$$

From each matrix $\overline{\mathbf{T}}_i$ we obtain a parametrization of a subset of S_6^2 depending on four parameters. This subset essentially describes the motion capability of one limb with fixed active prismatic joint. The final step is then to eliminate from each parametrization the corresponding four joint parameters $t_{i1},t_{i2},t_{i3},t_{i4}$ to obtain algebraic equations which contain only the Study parameters and, of course, the design parameters h_1 and h_2 and the active motion parameters r_1, r_2, r_3. This can be achieved easily by using the Linear Implicitization Algorithm (LIA).

To check the result another method was used to deduce these equations. In contrast to the systematic way from before, this method is based on geometric insight and for more complex link combinations it is quite likely that it doesn't work. Aside from Eq. (5) two other equations have to be deduced for each leg. As it can be seen in Fig. 19 the distance of point $\mathbf{A}_i$ and $\mathbf{B}_i$ always remains constant r_i. By describing both points with respect to Σ_0 using matrices Eq.(38) and Eq.(39) and transformations $\mathbf{F}_{ij}$ the Euclidean distance $|\mathbf{A}_i\mathbf{B}_i|^2$ can be computed and equated with r_i^2. After removal of denominators an equation of degree four in $x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3$ remains. After applying a trick which was used by Husty (1996) the degree can be lowered to two. The trick is to add a multiple of the square of Eq. (5) to obtain an expression with the factor Eq.(6), which can immediately be removed because it will never vanish. All equations generated this way appear also in the result of the previous method. Note that these equations was also derived in Example 11 Eq.(53) on page 33. The other condition is that the first and last axis in each leg are constrained to stay coplanar. By using points $\mathbf{A}_i$ and $\mathbf{B}_i$ from above and another two points lying on the axes a 4×4 matrix and its determinant can be computed. This determinant is the condition for coplanarity and has again degree four. It is possible to lower the degree to two by subtracting a proper multiple of Eq. (5) and removing the emerging factor Eq.(6). The resulting equations are

again identical to those derived by the LIA. After some simplifications of all seven equations to remove $\sqrt{3}$ from at least some equations the following equations are obtained:

$$
\begin{aligned}
&g_1 : x_0 y_0 + x_1 y_1 + x_2 y_2 + x_3 y_3 = 0 \\
&g_2 : (h_1 - h_2) x_0 x_2 + (h_1 + h_2) x_1 x_3 - x_2 y_3 - x_3 y_2 = 0 \\
&g_3 : (h_1 - h_2) x_0 x_3 - (h_1 + h_2) x_1 x_2 - 4 x_1 y_1 - 3 x_2 y_2 - x_3 y_3 = 0 \\
&g_4 : (h_1 - h_2) x_0 x_3 - (h_1 + h_2) x_1 x_2 + 2 x_1 y_1 + 2 x_3 y_3 = 0 \\
&g_5 : (h_1^2 - 2 h_1 h_2 + h_2^2 - r_1^2) x_0^2 + 2\sqrt{3}(h_1 - h_2) x_0 y_2 - 2 (h_1 - h_2) x_0 y_3 + \\
&\quad (h_1^2 + 2 h_1 h_2 + h_2^2 - r_1^2) x_1^2 - 2 (h_1 + h_2) x_1 y_2 - 2\sqrt{3}(h_1 + h_2) x_1 y_3 + \\
&\quad (h_1^2 - h_1 h_2 + h_2^2 - r_1^2) x_2^2 + 2\sqrt{3} h_1 h_2 x_2 x_3 - 2\sqrt{3}(h_1 - h_2) x_2 y_0 + \\
&\quad 2 (h_1 + h_2) x_2 y_1 + (h_1^2 + h_1 h_2 + h_2^2 - r_1^2) x_3^2 + 2 (h_1 - h_2) x_3 y_0 + \\
&\quad 2\sqrt{3}(h_1 + h_2) x_3 y_1 + 4 (y_0^2 + y_1^2 + y_2^2 + y_3^2) = 0 \\
&g_6 : (h_1^2 - 2 h_1 h_2 + h_2^2 - r_2^2) x_0^2 - 2\sqrt{3}(h_1 - h_2) x_0 y_2 - 2 (h_1 - h_2) x_0 y_3 + \\
&\quad (h_1^2 + 2 h_1 h_2 + h_2^2 - r_2^2) x_1^2 - 2 (h_1 + h_2) x_1 y_2 + 2\sqrt{3}(h_1 + h_2) x_1 y_3 + \\
&\quad (h_1^2 - h_1 h_2 + h_2^2 - r_2^2) x_2^2 - 2\sqrt{3} h_1 h_2 x_2 x_3 + 2\sqrt{3}(h_1 - h_2) x_2 y_0 + \\
&\quad 2 (h_1 + h_2) x_2 y_1 + (h_1^2 + h_1 h_2 + h_2^2 - r_2^2) x_3^2 + 2 (h_1 - h_2) x_3 y_0 - \\
&\quad 2\sqrt{3}(h_1 + h_2) x_3 y_1 + 4 (y_0^2 + y_1^2 + y_2^2 + y_3^2) = 0 \\
&g_7 : (h_1^2 - 2 h_1 h_2 + h_2^2 - r_3^2) x_0^2 + 4 (h_1 - h_2) x_0 y_3 + (h_1^2 + 2 h_1 h_2 + h_2^2 - r_3^2) x_1^2 + \\
&\quad 4 (h_1 + h_2) x_1 y_2 + (h_1^2 + 2 h_1 h_2 + h_2^2 - r_3^2) x_2^2 - 4 (h_1 + h_2) x_2 y_1 + \\
&\quad (h_1^2 - 2 h_1 h_2 + h_2^2 - r_3^2) x_3^2 - 4 (h_1 - h_2) x_3 y_0 + 4 (y_0^2 + y_1^2 + y_2^2 + y_3^2) = 0
\end{aligned}
\tag{60}
$$

This system of algebraic equations describes the mechanism. Fixing the motion parameters r_i one can ask for all projective points in $\mathbb{P}^7$ which fulfill all these seven equations, under the condition that $x_0^2 + x_1^2 + x_2^2 + x_3^2 \neq 0$. These points represent then all possible poses of the platform (assemblies of the manipulator) which are the solution of the direct kinematics of the TSAI 3-UPU manipulator for given fixed inputs. Because it is more convenient to do all computations in affine space, without loss of generality, the following normalization equation is added:

$$
g_8 : \; x_0^2 + x_1^2 + x_2^2 + x_3^2 - 1 = 0 \tag{61}
$$

Now it is guaranteed that no solution of this final system lies in the forbidden variety described by $x_0^2 + x_1^2 + x_2^2 + x_3^2 = 0$ (see Eq. (6)). The downside of the normalization is that for each projective solution point two affine representatives as solutions for Eqs. (60)-(61) are obtained. This has to be taken in account when different solutions are counted.

5.4 Solving the system

In the previous subsection a set of eight algebraic equations was deduced whose solutions describe all poses of the TSAI 3-UPU mechanism, depending on fixed parameters h_1 and h_2 and of course the active joint parameters r_1, r_2, and r_3. In the following this system of equations is always written as a polynomial ideal in the variables $x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3$ over the coefficient ring $\mathbb{C}[h_1, h_2, r_1, r_2, r_3]$. The ideal that has to be dealt with is

$$\mathscr{J} = \langle g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8 \rangle,$$

where each g_i here denotes the polynomial on the left hand side of the corresponding constraint equation. To begin with the following ideal $\mathscr{J}$ is inspected, which is obviously independent of the joint parameters r_1, r_2, and r_3

$$\mathscr{J} = \langle g_1, g_2, g_3, g_4 \rangle.$$

Allthough that ideal is not that complicated, it is checked if it is possible to split $\mathscr{J}$ into smaller ideals such that $\mathscr{J}$ can be written as

$$\mathscr{J} = \bigcap_i \mathscr{J}_i.$$

The benefit of such a decomposition would be that the vanishing set $\mathscr{V}(J)$ of $\mathscr{J}$ would be the union of the vanishing sets (i.e. varieties) of the ideals $\mathscr{J}_i$. As a simple example one could imagine a polynomial equation which can be written as product of several factors, i.e. the variety of the equation is a union of simpler varieties. This concept can be extended to arbitrary ideals and is called *primary (prime) decomposition*, where the difference between *prime* and *primary* lies only in the way how the resulting ideals are returned. *Prime decomposition* is the more strict version and returns all ideals reduced down to the bare essentials (the results are *radicals*). Relating to the example mentioned above *primary decomposition* would return the factors with possible powers, whereas *prime decomposition* only the factors with multiplicity one (see Section 4).

The computation of the prime decomposition of $\mathscr{J}$ shows that it can indeed be written in a very simple way:

$$\mathscr{J} = \bigcap_{i=1}^{6} \mathscr{J}_i$$

with

$$\mathscr{J}_1 = \langle y_0, x_1, x_2, x_3 \rangle, \ \mathscr{J}_2 = \langle x_0, y_1, x_2, x_3 \rangle,$$
$$\mathscr{J}_3 = \langle y_0, y_1, x_2, x_3 \rangle, \ \mathscr{J}_4 = \langle x_0, x_1, y_2, y_3 \rangle,$$

$$\mathscr{J}_5 = \langle (h_1-h_2)x_0x_2+(h_1+h_2)x_1x_3-x_2y_3-x_3y_2, (h_1-h_2)x_0x_3- (h_1+h_2)x_1x_2-x_2y_2+x_3y_3, 2x_1y_1+x_2y_2+x_3y_3, x_0y_0-x_1y_1, (h_1-h_2)^2x_0^2+(h_1+h_2)^2x_1^2-y_2^2-y_3^2, (h_1+h_2)x_2^3y_0-3(h_1-h_2)x_2^2x_3y_1- 2x_2^2y_0y_1-3(h_1-h_2)x_2x_3^2y_0+(h_1-h_2)x_3^3y_1-2x_3^2y_0y_1\rangle$$

$$\mathscr{J}_6 = \langle x_0, x_1, x_2, x_3\rangle.$$

This primary decomposition was computed over the ring $\mathbb{Q}[x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, h_1, h_2]$ to find possible other decompositions or changes of dimension, which could occur for special values of h_1 and h_2. Fortunately over this ring ideal $\mathscr{J}_5$ does not split and has always the same dimension, i.e. dimension 3. Although ideal $\mathscr{J}_6$ is a valid component of $\mathscr{J}$ it can be removed, because in connection with Eq. (6) it leads to an inconsistent system. For the zero set or vanishing set $\mathscr{V}(J)$ of $\mathscr{J}$ it follows that

$$\mathscr{V}(J) = \bigcup_{i=1}^{5} \mathscr{V}(\mathscr{J}_i).$$

Now the remaining equations are added and by generating the sub-systems

$$\mathscr{K}_i := \mathscr{J}_i \cup \langle g_5, g_6, g_7, g_8\rangle$$

the vanishing set of the whole system $\mathscr{I}$ can be written as

$$\mathscr{V}(I) = \bigcup_{i=1}^{5} \mathscr{V}(\mathscr{K}_i).$$

So, instead of studying the system as a whole, each of the smaller sub-systems $\mathscr{K}_i$ can be inspected separately. Then the solution of system $\mathscr{I}$ is the union of the solutions of the sub-systems.
Furthermore, it has to be noted that for each assembly mode described by a solution of direct kinematics, there exists another solution where the manipulator is assembled mirrored with respect to the base.

Solutions for arbitrary design parameters In the following subsection all computations for direct kinematics are performed under the assumption that the five parameters are arbitrary i.e. generic. To find out the Hilbert dimension of each ideal $\mathscr{K}_i$ the necessary Groebner bases are computed for general parameters, except the basis for $\mathscr{K}_5$, where for reasons of simplification randomly chosen parameters h_1, h_2 are substituted before the computation of the basis. So, for arbitrary design parameters the result is that

$$dim(\mathscr{K}_i) = 0, \;\; i = 1, \ldots, 5$$

which means that all sub-systems have finitely many solutions. Reusing the computed bases from above the number of solutions can be determined easily for each system $\mathscr{K}_i$. Due to the fact that always two solutions of a system describe the same position of the platform, each of these numbers has to be halved (see paragraph below Eq. (61)). In the following only these essentially different solutions are considered. The number of solutions for each system $\mathscr{K}_i$ in the generic case is

Table 3. Essentially different solutions for each ideal $\mathscr{K}_1, \ldots, \mathscr{K}_5$

$\mathscr{K}_1$	$\mathscr{K}_2$	$\mathscr{K}_3$	$\mathscr{K}_4$	$\mathscr{K}_5$	total
2	2	4	6	64	78

So all together there are 78 essentially different solutions for given generic parameters h_1, h_2 and joint parameters r_1, r_2, and r_3, i.e. 78 possible poses of the platform, theoretically. It is clear that for arbitrarily chosen parameters all these solutions will be complex. Mechanically this means that the manipulator cannot be assembled because of e.g. too different limb lengths. But of course, when design and joint parameters are chosen thoughtfully, many of the solution poses also can be real. And, as already mentioned, if there are real solutions at all, they appear in pairs where one pose can be assembled "upwards", the other one downwards. In Section 5.5 design parameters will be given for almost all $\mathscr{K}_i$ such that all solutions of the corresponding system are real. Due to the fact that systems $\mathscr{K}_1, \mathscr{K}_2, \mathscr{K}_3$ allow a closed form solution it can be shown that it is not possible that all systems can reach their maximum of real solutions simultaneously. To get an idea what the total number of real solutions for the manipulator can be some attempts were made using reasonably chosen parameters. Surprisingly this number never exceeded 28 in our computations. According to private correspondence with Damien Chablat he found a set of parameters where the number of essentially different solutions is 32. But it is absolutely not clear if this is the upper bound for the number of real solutions.

Solutions for equal limb lengths Here it is assumed that all limbs are of equal length. Because of the structure of the equations this is a non-generic case and has to be treated separately.

$$r_1 := r \quad r_2 := r \quad r_3 := r.$$

Now the same computations can be performed which were done in the previous subsection to obtain the Hilbert dimension of each ideal. Due to the fact that there are less parameters all Groebner bases can be computed without specifying any

parameters. For the dimension the result is the same as in the previous case.

$$dim(\mathscr{K}_i) = 0, \;\; i = 1,\ldots,5.$$

When the number of solutions is computed for each system and halved afterwards the following results are obtained.

Table 4. Essentially different solutions, limb lengths are equal

$\mathscr{K}_1$	$\mathscr{K}_2$	$\mathscr{K}_3$	$\mathscr{K}_4$	$\mathscr{K}_5$	total
2	2	2	6	60	72

Here there exist theoretically 72 solutions for the platform's position, six less than before. Concerning the question where they could have gone, one should not forget that there is this forbidden subvariety on the Study-quadric. It is possible that for special design parameters a solution lies on this subvariety. In the previous paragraph it was mentioned that for special design parameters 28 real solutions can be obtained. Actually it was a set of parameters with equal limb lengths. The exact values for the parameters were

$$h_1 = 12, h_2 = 7, r_1 = r_2 = r_3 = \frac{181}{13} \approx 13.923.$$

A very important difference to the SNU 3-UPU manipulator which was discussed in Walter et al. (2009) is the fact that here in general all 72 solutions have multiplicity 1, i.e. particularly the solution which corresponds to the so called "home position" has multiplicity 1, and not 4, which is the case for the SNU 3-UPU. One could conclude that the TSAI 3-UPU should show a better behavior in the home position, not being that shaky as the SNU 3-UPU. According to the results from Dwarakanath and Bhutani (2012) this is absolutely the case, provided that the mechanism is built with minimal clearences in all joints. The home pose can be seen in Fig. 19 and is described by the following values

$$x_0 = 1, \;\; x_1 = 0, \;\; x_2 = 0, \;\; x_3 = 0,$$

$$y_0 = 0, \;\; y_1 = \sqrt{r^2 - (h_1 - h_2)^2}/2, \;\; y_2 = 0, \;\; y_3 = 0.$$

5.5 The manipulator's operation modes

Until now r_1, r_2, and r_3 were treated as fixed design parameters for the direct kinematics. In this section they will be treated as parameters which are allowed to change, i.e. the behavior of this mechanism will be studied, when the prismatic

joints are actuated. Computation of the Hilbert dimension of each ideal $\mathscr{K}_i$ with r_1, r_2, r_3 used as unknowns shows that

$$\overline{dim}(\mathscr{K}_i) = 3, \;\; i = 1, \ldots, 5$$

where $\overline{dim}$ denotes the dimension over $\mathbb{C}[h_1, h_2]$, in contrast to dim which denotes the dimension over $\mathbb{C}[h_1, h_2, r_1, r_2, r_3]$ as in the previous sections. It follows that for generic design parameters the 3-UPU manipulator has 3 DOFs.
As it was shown in Walter et al. (2009) for the SNU 3-UPU manipulator each subsystem $\mathscr{K}_i$ of a mechanism's set of equations corresponds to a specific operation mode of the manipulator. In the following each system $\mathscr{K}_i$ will be discussed separately. Possible singularities which can emerge either as singular poses of a mode itself, or as poses where a change of modes is feasible, will be discussed in the next sections.
The following algorithm is applied in the next paragraphs: Each system $\mathscr{J}_i$ is solved and the solution is substituted into the transformation matrix $\mathbf{M}$ Eq.(3). From the obtained results properties of the solutions of the sub-systems $\mathscr{K}_i$ can be deduced and from these follow properties of the motion of the platform. It is absolutely not necessary to use equations $g_5 - g_7$ from Eq.(60) for this inspection, because they describe only the limb lengths which are now free parameters. Equation (61) is used to simplify the matrix entries, if possible, i.e. if three unknowns of Eq.(61) are 0, the remaining unknown is set to 1. Furthermore the position of the platform is described by a series of simpler transformations, starting from the "planar home position".

System $\mathscr{K}_1$: Translational Mode

$$\{y_0 = 0, x_1 = 0, x_2 = 0, x_3 = 0\}$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2y_1 & 1 & 0 & 0 \\ -2y_2 & 0 & 1 & 0 \\ -2y_3 & 0 & 0 & 1 \end{pmatrix}.$$

This is the simple translational operation mode which was discussed in many papers about this manipulator. The transformation matrix was simplified by substituting $x_0 = 1$. From the matrix it can easily be seen that the translational motion can be parameterized using y_1, y_2, y_3. If r_1, r_2, r_3 are given this mode leads to two assembly modes (see Tab. 3) and both are real e.g. for the parameters

$$h_1 = 12, h_2 = 7, r_1 = \frac{670}{21}, r_2 = \frac{243}{7}, r_3 = \frac{611}{21}.$$

System $\mathscr{K}_2$: Twisted translational Mode

$$\{x_0 = 0, y_1 = 0, x_2 = 0, x_3 = 0\}$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2y_0 & 1 & 0 & 0 \\ 2y_3 & 0 & -1 & 0 \\ -2y_2 & 0 & 0 & -1 \end{pmatrix}.$$

This is a mode which was already mentioned in Chebbi and Parenti-Castelli (2010), where singularities of different geometries of the 3-UPU translational manipulator are investigated. The twisted translational mode appears in this paper as geometry B.1. Here $x_1 = 1$ was used to simplify the result. Each solution of system $\mathscr{K}_2$ corresponds to a rotation of the platform about its normal axis N by 180 degrees and a subsequent translation. It follows that the described operation mode is basically a pure translation. To parameterize this motion one could use y_0, y_2, y_3 as parameters. All solutions for direct kinematics are real for e.g. the parameters

$$h_1 = 12, h_2 = 7, r_1 = \frac{121}{3}, r_2 = \frac{965}{21}, r_3 = \frac{62}{3}.$$

System $\mathscr{K}_3$: Planar Mode

$$\{y_0 = 0, y_1 = 0, x_2 = 0, x_3 = 0\}$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2(-x_0 y_2 + x_1 y_3) & 0 & x_0^2 - x_1^2 & -2x_0 x_1 \\ 2(-x_0 y_3 - x_1 y_2) & 0 & 2x_0 x_1 & x_0^2 - x_1^2 \end{pmatrix}.$$

Solutions of $\mathscr{K}_3$ correspond to poses of the platform where it is coplanar to the base. To parameterize this planar operation mode one could use x_1, y_2, y_3 in connection with $x_0^2 + x_1^2 = 1$, where x_1 is responsible for the rotation of the platform about its normal axis N and y_2, y_3 for the translation in the base-plane, i.e free planar motion is available in this mode. Concerning assembly modes belonging to this mode all four solutions of direct kinematics are real for e.g. the parameters

$$h_1 = 12, h_2 = 7, r_1 = \frac{125}{7}, r_2 = \frac{493}{21}, r_3 = \frac{66}{7}.$$

System $\mathscr{K}_4$: Upside-down planar mode

$$\{x_0 = 0, x_1 = 0, y_2 = 0, y_3 = 0\}$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 2(x_2 y_0 - x_3 y_1) & 0 & x_2^2 - x_3^2 & 2x_2 x_3 \\ 2(x_2 y_1 + x_3 y_0) & 0 & 2x_2 x_3 & -x_2^2 + x_3^2 \end{pmatrix}.$$

Solutions of $\mathscr{K}_4$ correspond to positions of the platform where it is turned upside down and coplanar to the base. This can be achieved by starting in the planar home position with disconnected legs, turning the platform about its y-axis by 180 degrees, and attaching the legs again. To parameterize the upside-down planar operation mode one could use x_3, y_0, y_1 in connection with $x_2^2 + x_3^2 = 1$, where x_3 is responsible for the rotation of the platform about its normal axis N and y_0, y_1 for the translation in the base-plane. It was possible to find parameters where all six solutions of direct kinematics are real, which are e.g.

$$h_1 = 12, h_2 = 7, r_1 = \frac{125}{7}, r_2 = \frac{493}{21}, r_3 = \frac{66}{7}.$$

System $\mathscr{K}_5$: General Mode

This mode is the most difficult one, because of the complexity of the equations in ideal $\mathscr{J}_5$. What definitely can be said is that it has in general for given limb length 64 solutions and that the system has 28 real solutions if the parameters are

$$h_1 = 12, h_2 = 7, r_1 = r_2 = r_3 = \frac{181}{13}.$$

Although the equations of $\mathscr{J}_5$ can be solved linearly for x_0, x_1,y_2, and y_3, due to the complexity of the equations there is no possibility to find a neat description of this mode, neither the condition for singularities nor a description of them.
Summing up there are five essentially different operation modes of the TSAI 3-UPU manipulator. It is quite obvious that this mechanism is more complex than the SNU 3-UPU whose essentially seven operation modes were quite simple.

5.6 Conditions for singular poses

In this section singular poses of the TSAI 3-UPU manipulator are discussed. A singular pose is a pose where the Jacobian matrix of Eqns. (60)-(61) with respect to the unknowns $x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3$ has vanishing determinant. If this determinant is computed a complete description of all possible forward singular poses is available. Because the manipulator is actuated via prismatic joints there are no input singularities in the kinematic chains of the legs. Due to the fact that the manipulator has more than one operation mode, singular poses can be divided into two different kinds. Either such a pose belongs to a single operation mode, or it is a pose which becomes singular because it lies in more than one operation

mode. The latter is then a pose where the manipulator can change its operation mode. Seen from the point of ideals the ideal $\mathscr{I}$ describes a three dimensional variety in $\mathbb{C}[h_1,h_2][x_0,x_1,x_2,x_3,y_0,y_1,y_2,y_3,r_1,r_2,r_3]$, which is composed of five simpler varieties. Singular poses lie now either on one of these components or in an intersection, which can be at most two dimensional.
In this subsection necessary conditions on the lengths r_1,r_2,r_2 are deduced for obtaining singular poses of the first kind. Because these poses always correspond to a single operation mode it is possible to use ideals $\mathscr{K}_i$ instead of $\mathscr{I}$. For almost all ideals the procedure is the same:
The 8×8 Jacobian matrix and its determinant are computed. This equation is added to the ideal $\mathscr{K}_i$ and from the resulting ideal all variables x_0,x_1,x_2,x_3,y_0,y_1, y_2,y_3 are eliminated. The result is an equation in $\mathbb{Q}[h_1,h_2][r_1,r_2,r_3]$ which is the necessary condition that the direct kinematics of the manipulator leads to a singular pose in this mode. The ideal for the *General Mode* needs a special treatment because it is too complex for the standard elimination methods.
Due to the fact that leg lengths r_1,r_2,r_2 appear squared in the equations the result of the elimination describes a hypersurface in the space of these parameters, which has full octahedral symmetry. The surface looks the same in all octants and therefore only the first octant is shown in the following figures.

Singular poses: Translational Mode The condition is an equation of degree 4 in r_1,r_2,r_3 and looks as follows:

$$r_1^4+r_2^4+r_3^4-r_1^2r_2^2-r_1^2r_3^2-r_2^2r_3^2-3\,(h_1-h_2)^2\,(r_1^2+r_2^2+r_3^2)+9\,(h_1-h_2)^4=0. \tag{62}$$

Further inspection shows that for given values for h_1 and h_2 Eq.(62) describes a Kummer surface in the joint parameter space r_1,r_2,r_3. The Kummer surface is a very famous algebraic degree four surface, possessing the maximum number of 16 double points (see e.g. Hudson (1905)). If the joint parameters r_1,r_2,r_3 are chosen inside resp. outside the "pipe" (Fig. 20), both solutions of the direct kinematics are real resp. complex. On the other hand, if the parameters of the limbs are chosen such that the condition Eq.(62) is fulfilled, it is not difficult to show that both solutions coincide and the corresponding positions are described by

$$\mathbf{M}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\p_2&0&1&0\\p_3&0&0&1\end{pmatrix},$$

i.e. the singular positions of the translational mode are those, where base and platform lie in the same plane.

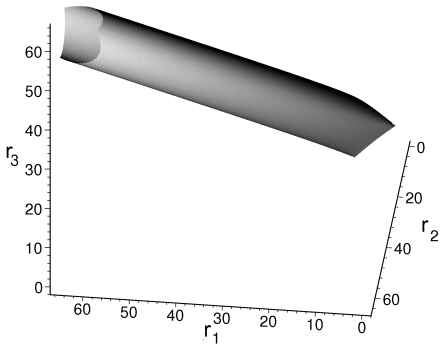


Figure 20. The singularity surface of the translational mode.

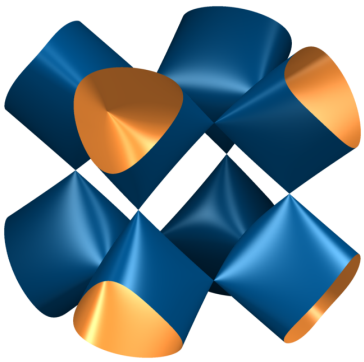


Figure 21. Kummer singularity surface in all eight octants

Singular poses: Twisted translational Mode The necessary condition for singular positions:

$$r_1^4 + r_2^4 + r_3^4 - r_1^2 r_2^2 - r_1^2 r_3^2 - r_2^2 r_3^2 - 3\,(h_1 + h_2)^2\,(r_1^2 + r_2^2 + r_3^2) + 9\,(h_1 + h_2)^4 = 0 \tag{63}$$

It is again a Kummer surface with the difference that the ”pipe“ (Fig. 22) has a larger diameter. Again, if the limb parameters are chosen inside resp. outside the “pipe“, both solutions of the direct kinematics are real resp. complex. If the condition Eq.(63) is fulfilled, both solutions coincide and the corresponding positions

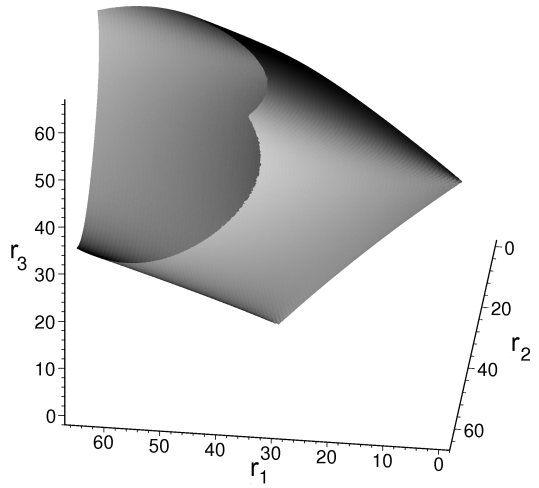


Figure 22. The singularity surface of the twisted translational mode.

are described by

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ p_2 & 0 & -1 & 0 \\ p_3 & 0 & 0 & -1 \end{pmatrix},$$

i.e. the singular positions of the twisted translational mode are those, where base and platform lie in the same plane and the platform is rotated by 180 degrees in this plane.

Singular poses: Planar Mode The condition is an equation of degree 12 which can be factorized into

$$F_1 F_2 (r_1 + r_2 - r_3)(r_1 + r_3 - r_2)(r_2 + r_3 - r_1) F_3 = 0 \tag{64}$$

where F_1 and F_2 are exactly the singularity conditions from the previous modes and F_3 is the factor $r_1 + r_2 + r_3$ which does never vanish. The union of the corresponding varieties of these five factors separates the joint parameter space into several zones where zero, two, or four real solutions are obtained. Due to the fact that the condition is a product of essentially five factors there are more than one representatives for a singular position. First of all both matrices from the previous modes, describing their singular positions. Furthermore from the small factors one

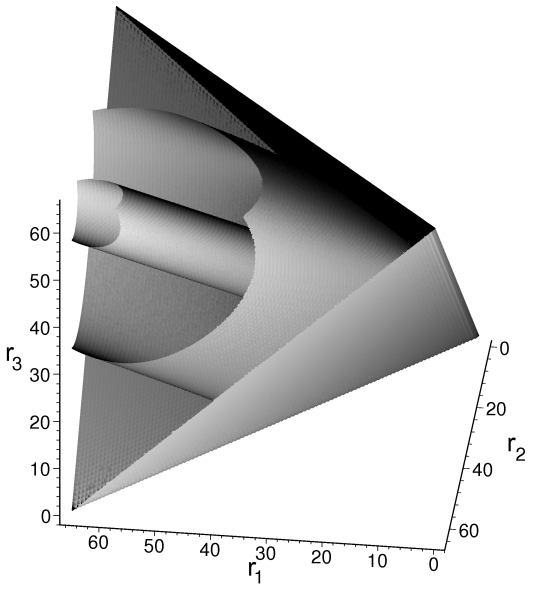


Figure 23. The singularity surface of the planar mode. It contains the surfaces from Fig. 20 and Fig. 22.

obtains

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ p_2 & 0 & v_{33} & v_{32} \\ p_3 & 0 & -v_{32} & v_{33} \end{pmatrix}$$

with

$$v_{32}^2 + v_{33}^2 = 1, \quad p_2^2 + p_3^2 = h_1^2 + h_2^2 - 2\,h_1\,h_2\,v_{33},$$

i.e. amongst others a singular position is obtained when the origin of the platform frame lies on a circle in the base plane whose radius is determined by the rotation of the frame about its x-axis.

Singular poses: Upside-down planar mode Computation of the singularity condition was rather hard in this case. The result is a very lengthy equation of degree 24 which cannot be factorized over $\mathbb{Q}$. Due to space limitations it is not displayed here. Several plots of it were made for given h_1, h_2. The variety is again tube-like but it has also self-intersections which again separate the parameter space in different zones. It is not clear how the singular poses arising from this condition can be described in a short and concise way.

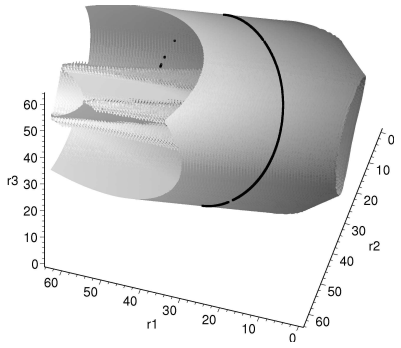


Figure 24. The singularity surface of the upside-down planar mode.

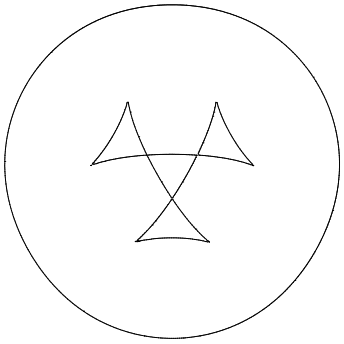


Figure 25. Intersection curve of the "pipe" of Fig. 24.

Singular poses: General Mode Although the equations of $\mathscr{J}_5$ can be solved linearly for x_0, x_1,y_2, and y_3, due to the complexity of the equations there is no possibility to find a neat description of this mode. Surprisingly the condition for singularities could be computed for given values of h_1 and h_2, but in a very different way. First of all the two design parameters are set to

$$h_1 = 12, h_2 = 7.$$

The ideal describing all singular poses of this mode is computed in an indirect way. The reason for this is the fact that the number of describing equations is not eight as in the previous cases. To begin with, the Jacobian determinant is derived from Eqns. (60)-(61) and it is added to the ideal $\mathscr{I}$. This ideal has dimension 2 and describes on one hand all singular poses of all modes and on the other hand all singular poses arising of intersections of components. But especially it is again a composition of various varieties. All components of this variety which are not of interest are described by ideals which contain an ideal $\mathscr{J}_i$ with $i = 1,\ldots,4$. To obtain the desired component all superfluous components have to be removed. Essentially this can be achieved by the technique of *saturation* with respect to each of the polynomials x_0,x_1,x_2,x_3,y_0 and y_1, which are the generating elements of ideals $\mathscr{J}_1,\ldots,\mathscr{J}_4$. By inspection of the intersection of the variety with each of these hyperplanes it is checked that the only two dimensional components where one of these variables is zero, are the superfluous ones. Saturation of an ideal with respect to a polynomial (which can be seen as a principal ideal) is nothing else then the computation of their quotient for multiple times until the result does not change anymore. A possible analogy would be the removal of a power of a prime number from a composite number by repeated division. The result is an ideal $\mathscr{L}$

which describes all singular poses of the general mode.
The tricky part is the elimination of the Study parameters. Due to the fact that all present methods and algorithms fail for reasons of memory and time, another way has to be found. First of all randomly chosen parameters are substituted for r_1 and r_2. Then the elimination can be done and the result is a univariate polynomial in r_3 of degree 208 (only even powers). From this it is concluded that the equation wanted has in general degree 208 in each unknown. For the total degree one has to guess and the following computations show that a guess of also 208 is correct. Then a general ansatz for a polynomial in r_1, r_2, r_3 of degree 208 is made, introducing 34497 unknown coefficients $C_1, \ldots, C_{34497}$ which have to be determined. Now the following steps are repeated for several pairs of values for r_1 and r_2 (for 1413 pairs to be exact): At first the values of the pair are substituted into ideal $\mathscr{L}$ and the univariate polynomial in r_3 is obtained. Then the substitution is made in the general ansatz, leading to a polynomial in r_3 where the coefficients are linear expressions in the coefficients C_i. After multiplication of the univariate with an unknown scaling factor comparison of coefficients leads to a system of 105 linear equations.
For each pair 105 such equations are obtained which means that in the end there are 148365 equations in 35910 unknowns. As a precaution and for reasons of linear dependencies the number of pairs of values is set so high. In the end all coefficients $C_1, \ldots, C_{34497}$ can be computed up to a scaling factor and after substitution into the general ansatz the condition for singular poses of the general mode is available. Due to the extreme size of this equation a 3D plot is not possible, but in a roundabout way a plot of a planar intersection of the surface in the first octant can be generated, like for the surface in Fig. 24 resp. Fig. 25. See Figure 26 for an impression of the complexity of this surface. Due to the fact that the surface has several connections to the parts in neighboring octants, there is no ”pipe“ shape any more.

5.7 Changing operation modes

As already mentioned there exist poses where the mechanism can change from one mode into another mode. One of them is e.g. the “planar home position” where the mechanism can bifurcate into the planar mode or into the translational mode. In the following an overview is given of those poses where a mode-change is possible. This is done by inspecting each pair of ideals $\{\mathscr{K}_i, \mathscr{K}_j\}$ with respect to common solutions. For each pair the dimension of the intersection of the corresponding varieties is computed and the following results are obtained.
The numbers in Table 5 correspond to the dimension of the intersection variety. As it can easily be seen there are four possible combinations of operation modes which have no pose in common, to change from one to the other a detour has to

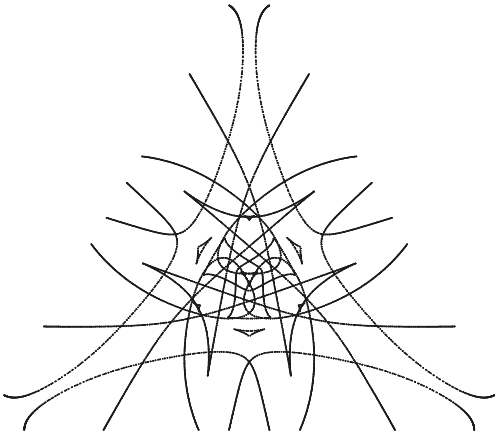


Figure 26. Intersection curve of the surface from the general mode, first octant, cut out by a plane orthogonal to $(1,1,1)^T$.

Table 5. All values of $\overline{dim}(\mathscr{K}_i \cup \mathscr{K}_j)$.

	$\mathscr{K}_1$	$\mathscr{K}_2$	$\mathscr{K}_3$	$\mathscr{K}_4$	$\mathscr{K}_5$
$\mathscr{K}_1$	...	-1	2	-1	2
$\mathscr{K}_2$	-1	...	2	-1	2
$\mathscr{K}_3$	2	2	...	-1	2
$\mathscr{K}_4$	-1	-1	-1	...	2
$\mathscr{K}_5$	2	2	2	2	...

be made, which is always possible via the general mode, corresponding to $\mathscr{K}_5$. Once more it has to be noted that mode changing poses of the manipulator are also singular poses, because each of them lies in the intersection of two varieties. In the following all possible changes are discussed with respect to necessary conditions and a description of related poses.

Translational mode ($\mathscr{K}_1$) $\longleftrightarrow$ planar mode ($\mathscr{K}_3$) The condition for this change is exactly the singularity condition from the translational mode, i.e. all singularities of this mode coincide with the intersection singularities with the planar mode. Hence the possible mode change poses have already been given in Subsection 5.6 and the corresponding singularity surface can be seen in Fig. 20.

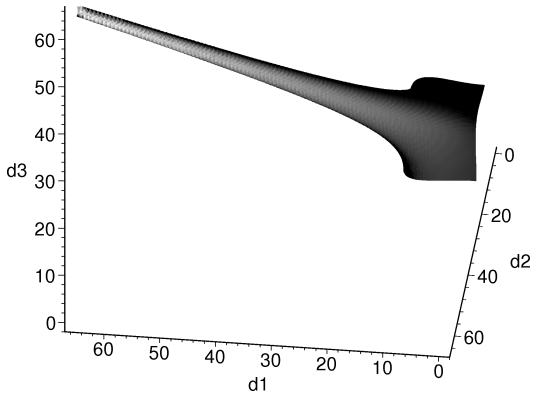


Figure 27. Surface describing the condition for changes from translational to general mode.

Translational mode ($\mathscr{K}_1$) $\longleftrightarrow$ general mode ($\mathscr{K}_5$) For this change a new condition appears, that is

$$r_1^4 + r_2^4 + r_3^4 - r_1^2 r_2^2 - r_1^2 r_3^2 - r_2^2 r_3^2 - 36\,(h_1 - h_2)^4 = 0 \tag{65}$$

and for the corresponding positions of the platform the following description can be deduced:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ p_1 & 1 & 0 & 0 \\ p_2 & 0 & 1 & 0 \\ p_3 & 0 & 0 & 1 \end{pmatrix}$$

with

$$p_2^2 + p_3^2 = 4\,(h_1 - h_2)^2,$$

i.e. platform and base have the same orientation and the origin of the platform's frame has to lie on a cylinder with radius $2\,(h_1 - h_2)$. This is a result which verifies statements about singularity loci from Chebbi and Parenti-Castelli (2010).

Twisted translational mode ($\mathscr{K}_2$) $\longleftrightarrow$ planar mode ($\mathscr{K}_3$) The condition for this change is exactly the singularity condition of the twisted translational mode, i.e. all singularities of this mode coincide with the intersection singularities with the planar mode. Hence the possible mode change poses have already been mentioned and the corresponding singularity surface can be seen in Fig. 22.

Twisted translational mode ($\mathscr{K}_2$) $\longleftrightarrow$ general mode ($\mathscr{K}_5$) The condition for this mode change is as follows:

$$r_1^4 + r_2^4 + r_3^4 - r_1^2 r_2^2 - r_1^2 r_3^2 - r_2^2 r_3^2 - 36\,(h_1 + h_2)^4 = 0. \tag{66}$$

For the corresponding poses of the platform the following description can be deduced:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ p_1 & 1 & 0 & 0 \\ p_2 & 0 & -1 & 0 \\ p_3 & 0 & 0 & -1 \end{pmatrix}$$

with

$$p_2^2 + p_3^2 = 4\,(h_1 + h_2)^2,$$

i.e. once more platform and base have the same orientation and the origin of the platform's frame has to lie on a cylinder, but this time with radius $2\,(h_1 + h_2)$. The same result can again be found in Chebbi and Parenti-Castelli (2010), p. 598.

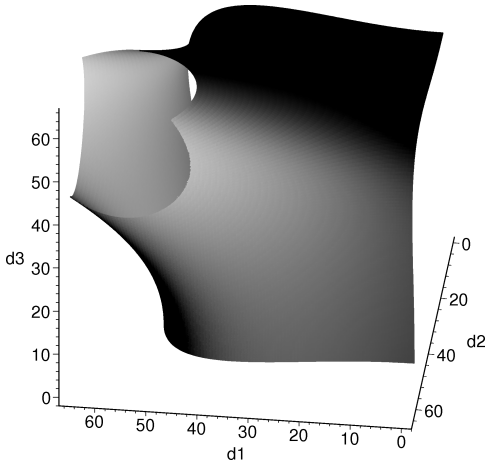


Figure 28. Surface describing the condition for changes from twisted translational mode to general mode.

Planar mode ($\mathscr{K}_3$) $\longleftrightarrow$ general mode ($\mathscr{K}_5$) The condition which has to be fulfilled for this case reads

$$7\,(r_1^4 + r_2^4 + r_3^4) - 11\,(r_1^2 r_2^2 - r_1^2 r_3^2 - r_2^2 r_3^2) = 0. \tag{67}$$

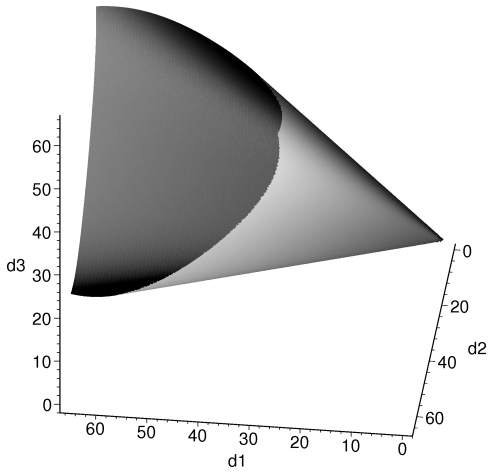


Figure 29. Surface describing the condition for changes from planar mode to general mode.

The corresponding poses of the platform are given by

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ p_2 & 0 & v_{33} & v_{32} \\ p_3 & 0 & -v_{32} & v_{33} \end{pmatrix}$$

with

$$v_{32}^2 + v_{33}^2 = 1, \quad p_2^2 + p_3^2 = 4\,(h_1^2 + h_2^2 - 2\,h_1\,h_2\,v_{33}).$$

It is noticeable that these positions are very similar to the singular poses of the planar mode itself.

Upside-down planar mode ($\mathscr{K}_4$) $\longleftrightarrow$ general mode ($\mathscr{K}_5$) The condition for this case can be computed but it is rather complicated. It is an equation of degree 24 and it is not equal to the singularity condition of the upside-down planar mode itself. As a result of the complexity up to now also no description of the platform's poses could be found.

All the other combinations which are $\{\mathscr{K}_1, \mathscr{K}_2\}$, $\{\mathscr{K}_1, \mathscr{K}_4\}$, $\{\mathscr{K}_2, \mathscr{K}_4\}$, and $\{\mathscr{K}_3, \mathscr{K}_4\}$ do not allow any operation mode change.

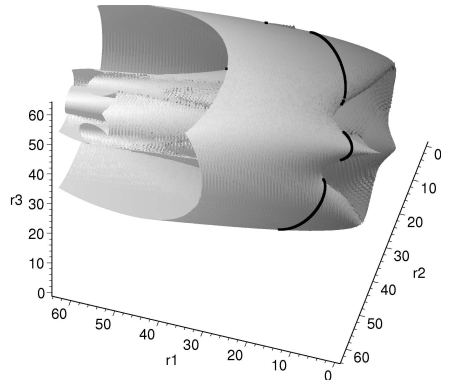


Figure 30. Surface describing the condition for changes from upside-down planar mode to general mode.

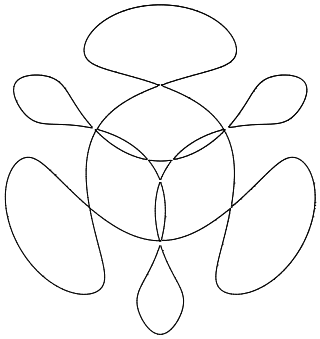


Figure 31. Intersection curve of the "pipe" of Fig. 30.

5.8 Conclusion

In this chapter an algebraic approach to the analysis of mechanisms was presented. To establish this methodology the Study representation of spatial displacements was introduced. Starting from the kinematic chain as basic element, different methods to derive the algebraic constraint equations for kinematic chains were demonstrated. After that, in a primer on methods from algebraic geometry, the most important and powerful algebraic tools for the analysis and solution of these equations were introduced. In the last section the application of all these methods was demonstrated in the complete global analysis of the 3-UPU TSAI parallel manipulator.

Bibliography

A. Angerer. *Rechentechnische Realisierung des inversen Roboterkinematik Problems*. PhD thesis, UMIT Hall, Tyrol,Austria, 2016.

M. Badescu, J. Morman, and C. Mavroidis. Workspace Optimization of 3-UPU Parallel Platforms with Joint Constraints. In *ICRA*, pages 3678–3683, 2002.

I. Bonev and D. Zlatanov. The Mystery of the Singular SNU Translational Parallel Robot, 2001. URL `http://www.parallemic.org/Reviews/Review004.html`.

I. A. Bonev. *Geometric Analysis Of Parallel Mechanisms*. Phd thesis, Université Laval, Québec, 2002.

O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland Publishing Company, 1979. ISBN 0-486-66346-9.

A. H. Chebbi. *The Potential of the 3-UPU Translational Parallel Manipulator and a Procedure to Select the Best Architecture*. PhD thesis, Università di Bologna, 2011.

A. H. Chebbi and V. Parenti-Castelli. Geometric and Manufacturing Issues of the 3-UPU Pure Translational Manipulator. In Doina Pisla, Marco Ceccarelli, Manfred Husty, and Burkhard Corves, editors, *New Trends in Mechanism Science: Analysis and Design*, pages 595–603. Springer, 2010.

D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 3 edition, 2007.

R. Di Gregorio and V. Parenti-Castelli. A Translational 3-DOF Parallel Manipulator. In Jadran Lenarčič and Manfred Husty, editors, *Advances in Robot Kinematics: Analysis and Control*, pages 49–58. Kluwer Academic Publishers, 1998.

R. Di Gregorio and V. Parenti-Castelli. Mobility Analysis of the 3-UPU Parallel Mechanism Assembled for a Pure Translational Motion. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta*, pages 520–525, 1999.

T. Dwarakanath and G. Bhutani. Novel Design Solution to High Precision 3 Axes Translational Parallel Mechanism. In S. Bandyopadhyay, G. Kumar, and P. Ramu, editors, *Machines and Mechanisms*, pages 373–381. Narosa Publishing House, New Dheli, 2012.

C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6(3):281–290, Jun 1990. ISSN 1042-296X. doi: 10.1109/70.56660.

Chanhee Han, Jinwook Kim, Jongwon Kim, and Frank Chongwoo Park. Kinematic sensitivity analysis of the 3-UPU parallel mechanism. *Mechanism and Machine Theory*, 37(8):787–798, 2002.

Bo Hu and Yi Lu. Solving stiffness and deformation of a 3-UPU parallel manipulator with one translation and two rotations. *Robotica*, 29(06):815–822, 2011.

R. W. H. T Hudson. *Kummer's Quartic Surface*. Camebridge University Press, 1905. reprint 1990 by Camebridge University Press.

M. L. Husty. An Algorithm for Solving the Direct Kinematics of General Stewart-Gough Platforms. *Mechanism and Machine Theory*, 31(4):365–380, 1996.

M. L. Husty, A. Karger, H. Sachs, and W. Steinhilper. *Kinematik und Robotik*. Springer, 1997.

M. L. Husty, M. Pfurner, and H.-P. Schröcker. A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mechanism and Machine Theory*, 42(1):66 – 81, 2007a. ISSN 0094-114X.

M. L. Husty, M. Pfurner, H.-P. Schröcker, and K. Brunnthaler. Algebraic methods in mechanism analysis and synthesis. *Robotica*, 25:661 – 675, 2007b.

M.L. Husty and H.-P. Schröcker. *21st Century Kinematics*, chapter Kinematics and Algebraic geometry, pages 85–123. Springer, 2012.

Ping Ji and Hongtao Wu. Kinematics analysis of an offset 3-UPU translational parallel robotic manipulator. *Robotics and Autonomous Systems*, 42(2):117–123, 2003.

A. Nayak, Th. Stigger, M. Husty, Ph. Wenger, and St. Caro. Operation Mode Analysis of 3-RPS Parallel Manipulators based on their Design Parameters. *Computer Aided Geometric Design*, 63: 122 – 134, July 2018. doi: 10.1016/j.cagd.2018.05.003. URL `https://hal.archives-ouvertes.fr/hal-01860789`.

L. Nurahmi, J. Schadlbauer, M. L. Husty, Ph. Wenger, and St. Caro. Kinematic analysis of the 3-RPS Cube Parallel Manipulator. *ASME Journ. of Mech. and Robotics*, 7(1):1—-11, 2015.

L. Nurahmi, St. Caro, Ph. Wenger, J. Schadlbauer, and M.L. Husty. Reconfiguration analysis of a 4-RUU parallel manipulator. *Mechanism and Machine Theory*, 96:269–289, 2016.

V. Parenti-Castelli, R. Di Gregorio, and F. Bubani. Workspace and Optimal Design of a Pure Translation Parallel Manipulator. *Meccanica*, 35(3):203–214, 2000.

M. Pfurner. *Analysis of spatial serial manipulators using kinematic mapping*. PhD thesis, University of Innsbruck, 2006. URL `http://repository.uibk.ac.at`.

M. Pfurner, H.-P. Schröcker, and M. L. Husty. Path planning in kinematic image space without the Study condition. In J.-P. Merlet and J. Lenarcic, editors, *Advances in Robot Kinematics*, pages https://hal.archives–ouvertes.fr/hal–01339423, 2016.

J. Schadlbauer. *Algebraic Methods in Kinematics and Line Geometry*. Doctoral thesis, University Innsbruck, 2014.

J. Schadlbauer, D. R Walter, and M. L. Husty. The 3-RPS parallel manipulator from an algebraic viewpoint. *Mechanism and Machine Theory*, 75:161–176, May 2014. ISSN 0094114X.

J. Schadlbauer, L. Nurahmi, M.L. Husty, Philippe Wenger, and St. Caro. *IAK: Proc. of the Intern. Conf., Lima, Peru, September 9-11, 2013*, chapter Operation Modes in Lower-Mobility Parallel Manipulators, pages 1–9. Springer International Publishing, 2015.

J. M. Selig. *Geometric Fundamentals of Robotics*. Springer Science+Business Media Inc., 2nd edition, 2005. ISBN 0-387-20874-7.

A. J. Sommese and Ch. W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005. ISBN 981-256-184-6.

Th. Stigger, A. Nayak, St. Caro, Ph. Wenger, M. Pfurner, and M.L. Husty. Algebraic Analysis of a 3-RUU Parallel Manipulator. *16th International Symposium on Advances in Robot Kinematics*, pages 141–149, 2018a.

Th. Stigger, M. Pfurner, and M.L. Husty. *Proceedings of the 7th European Conference on Mechanism Science*, volume 59 of *Mechanism and Machine Science*, chapter Workspace and Singularity Analysis of a 3-RUU Parallel Manipulator, pages 325 – 332. Springer, Dordrecht - Heidelberg - London - New York - Berlin, 2018b.

E. Study. *Geometrie der Dynamen*. Teubner, Leipzig, 1903.

Lung-Wen Tsai. Kinematics of a Three-DOF Platform with Three Extensible Limbs. In Jadran Lenarčič and Vincenzo Parenti-Castelli, editors, *Recent Advances in Robot Kinematics*, pages 401–410. Kluwer Academic Publishers, 1996.

Lung-Wen Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, Inc., 1999.

Lung-Wen Tsai and S. Joshi. Kinematics and Optimization of a Spatial 3-UPU Parallel Manipulator. *Journal of Mechanical Design*, 122:439–446, 2000.

S.-M. Varedi, H. M. Daniali, and D. D. Ganji. Kinematics of an offset 3-UPU translational parallel manipulator by the homotopy continuation method. *Nonlinear Analysis: Real World Applications*, 10(3):1767–1774, 2009.

D. R. Walter and M. L. Husty. On Implicitization of Kinematic Constraint Equations. In *Machine Design & Research (CCMMS 2010)*, volume 26, pages 218–226, Shanghai, 2010.

D. R. Walter, M. L. Husty, and Martin Pfurner. A Complete Kinematic Analysis of the SNU 3-UPU Parallel Robot. In Daniel Bates, GianMario Besana, Sandra Di Rocco, and Charles Wampler, editors, *Interactions of Classical and Numerical Algebraic Geometry*, volume 496 of *Contemporary Mathematics*, pages 331–346. American Mathematical Society, 2009.

A. Wolf, M. Shoham, and F. Chongwoo Park. An Investigation of the Singularities and Self-Motions of the 3-UPU Robot. In *Advances in Robot Kinematics*, pages 165–174. Kluwer Academic Publishers, 2002.

D. Zlatanov, R.G. Fenton, and B. Benhabib. Analysis of the instantaneous kinematics and singular configurations of hybrid-chain manipulators. In *Proceedings of the ASME* 23^{rd} *Biennial Mechanisms Conference*, volume DE-Vol. 72, page pp. 467–476, Minneapolis, MN, USA, September 11–14 1994.